Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

VISIT…

Studies in Computational Intelligence, Volume 149

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 127. Régie Gras, Einoshin Suzuki, Fabrice Guillet and Filippo Spagnolo (Eds.)
Statistical Implicative Analysis, 2008
ISBN 978-3-540-78982-6

Vol. 128. Fatos Xhafa and Ajith Abraham (Eds.)
Metaheuristics for Scheduling in Industrial and Manufacturing Applications, 2008
ISBN 978-3-540-78984-0

Vol. 129. Natalio Krasnogor, Giuseppe Nicosia, Mario Pavone and David Pelta (Eds.)
Nature Inspired Cooperative Strategies for Optimization (NICSO 2007), 2008
ISBN 978-3-540-78986-4

Vol. 130. Richi Nayak, Nikhil Ichalkaranje and Lakhmi C. Jain (Eds.)
Evolution of the Web in Artificial Intelligence Environments, 2008
ISBN 978-3-540-79139-3

Vol. 131. Roger Lee and Haeng-Kon Kim (Eds.)
Computer and Information Science, 2008
ISBN 978-3-540-79186-7

Vol. 132. Danil Prokhorov (Ed.)
Computational Intelligence in Automotive Applications, 2008
ISBN 978-3-540-79256-7

Vol. 133. Manuel Graña and Richard J. Duro (Eds.)
Computational Intelligence for Remote Sensing, 2008
ISBN 978-3-540-79352-6

Vol. 134. Ngoc Thanh Nguyen and Radoslaw Katarzyniak (Eds.)
New Challenges in Applied Intelligence Technologies, 2008
ISBN 978-3-540-79354-0

Vol. 135. Hsinchun Chen and Christopher C. Yang (Eds.)
Intelligence and Security Informatics, 2008
ISBN 978-3-540-69207-2

Vol. 136. Carlos Cotta, Marc Sevaux and Kenneth Sörensen (Eds.)
Adaptive and Multilevel Metaheuristics, 2008
ISBN 978-3-540-79437-0

Vol. 137. Lakhmi C. Jain, Mika Sato-Ilic, Maria Virvou, George A. Tsihrintzis, Valentina Emilia Balas and Canicious Abeynayake (Eds.)
Computational Intelligence Paradigms, 2008
ISBN 978-3-540-79473-8

Vol. 138. Bruno Apolloni, Witold Pedrycz, Simone Bassis and Dario Malchiodi
The Puzzle of Granular Computing, 2008
ISBN 978-3-540-79863-7

Vol. 139. Jan Drugowitsch
Design and Analysis of Learning Classifier Systems, 2008
ISBN 978-3-540-79865-1

Vol. 140. Nadia Magnenat-Thalmann, Lakhmi C. Jain and N. Ichalkaranje (Eds.)
New Advances in Virtual Humans, 2008
ISBN 978-3-540-79867-5

Vol. 141. Christa Sommerer, Lakhmi C. Jain and Laurent Mignonneau (Eds.)
The Art and Science of Interface and Interaction Design (Vol. 1), 2008
ISBN 978-3-540-79869-9

Vol. 142. George A. Tsihrintzis, Maria Virvou, Robert J. Howlett and Lakhmi C. Jain (Eds.)
New Directions in Intelligent Interactive Multimedia, 2008
ISBN 978-3-540-68126-7

Vol. 143. Uday K. Chakraborty (Ed.)
Advances in Differential Evolution, 2008
ISBN 978-3-540-68827-3

Vol. 144. Andreas Fink and Franz Rothlauf (Eds.)
Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management, 2008
ISBN 978-3-540-69024-5

Vol. 145. Mikhail Ju. Moshkov, Marcin Piliszczuk and Beata Zielosko
Partial Covers, Reducts and Decision Rules in Rough Sets, 2008
ISBN 978-3-540-69027-6

Vol. 146. Fatos Xhafa and Ajith Abraham (Eds.)
Metaheuristics for Scheduling in Distributed Computing Environments, 2008
ISBN 978-3-540-69260-7

Vol. 147. Oliver Kramer
Self-Adaptive Heuristics for Evolutionary Computation, 2008
ISBN 978-3-540-69280-5

Vol. 148. Philipp Limbourg
Dependability Modelling under Uncertainty, 2008
ISBN 978-3-540-69286-7

Vol. 149. Roger Lee (Ed.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2008
ISBN 978-3-540-70559-8

Roger Lee
(Ed.)

Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing

Springer

Prof. Roger Lee
Computer Science Department
Central Michigan University
Pearce Hall 413
Mt. Pleasant, MI 48859
USA
Email: lee1ry@cmich.edu

ISBN 978-3-540-70559-8 e-ISBN 978-3-540-70560-4

DOI 10.1007/978-3-540-70560-4

Studies in Computational Intelligence ISSN 1860949X

Library of Congress Control Number: 2008930269

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1
springer.com

Preface

The 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, held in Phuket Thailand on August 6–8, 2008 is aimed at bringing together researchers and scientist, businessmen and entrepreneurs, teachers and students to discuss the numerous fields of computer science, and to share ideas and information in a meaningful way. This publication captures 20 of the conference's most promising papers, and we impatiently await the important contributions that we know these authors will bring to the field.

In chapter 1, Chotirat Ann Ratanamahatana and Dechawut Wanichsan address the selection of optimal stopping criterion for semi-supervised learning methods for time series domains. The authors propose a novel stopping criterion and presents experimental results to support their conclusions.

In chapter 2, Jinyoung Ahn et al. address the emerging problem of spam in Voice over IP (VoIP) technology. The authors propose a fine tuning of the Progressive Multi Gray level approach to combating SPam over Internet Telephony (SPIT).

In chapter 3, Gihan Shin and Junchul Chun discuss facial expression recognition based on the Hidden Markov Model (HMM). The authors put forth an enhanced transition framework model for emotional state detection, and test its validity through experimentation.

In chapter 4, Mahmoud Jazzar and Aman Jantan propose an approach to anomaly intrusion detection based on causal knowledge reasoning. Using fuzzy cognitive maps, the authors present a method that attempts to diagnose and direct network traffic data based on its relevance to attack or normal connections.

In chapter 5, Belal Chowdhury et al. address challenges related with the deployment of an RFID-enabled system in the global supply chain. The paper outlines major RFID issues faced by supply chain management and presents a case study on pharmaceutical supply chain management applications by addressing and examining the issues of RFID implementation in an SCM system.

In chapter 6, Ho Min Sung et al. propose a clustering backup system that exploits file fingerprint mechanisms for multi-level duplication of redundant data. The authors' approach differs from the traditional file server system in two ways. First, they avoid data redundancy by making use of block-level fingerprinting; and second they apply clustering technology to reduce data transfer and I/O time to a fraction of a percent for each node. The authors conclude with experimentation that shows efficient use of storage space and noticeable performance improvement.

In chapter 7, Haeng Kon Kim presents a solution the problems associated with embedded software testing, whereby Aspect Oriented Programming is applied to simulate

test cases in the early phases of development. The author uses UML sequence diagrams to formalize sequences of events in order to generate test cases from use cases.

In chapter 8, Sinjae Lee et al. look into the field of Malicious Peers Behavior on peer-to-peer networks. The authors present an unprecedented analysis of the specific area of malicious behaviors of the Resource Chain Model.

In chapter 9, Montri Lawkobkit reports the results of a survey conducted in Bangkok, Thailand in 2007, and through these results presents a Thai perspective on Information Technology Service Management. The author concludes the article with recommendations for further research in the field.

In chapter 10, Rachid Elmeziane et al. propose a new artificial immune system with the unprecedented characteristic of unsupervised detection based on the mechanism of NK cell (Natural Killer cell). The authors apply the NK system, as it is dubbed, to the detection of fraud in mobile phones and present the results of the experiment.

In chapter 11, Takéhiko Nakama investigates the convergence properties of genetic algorithms (GAs) in 'noisy environments'. The author constructs and analyzes a Markov chain that modes the GAs in a novel approach to uncovering the convergence properties of the algorithms.

In chapter 12, Francoise Sailhan et al. propose a novel Hybrid Distributed Security Operation Center (HDSOC) to address security supervision for hybrid wired-cum-wireless networks. The system couples intrusion detection mechanisms with preventive measures in order to identify unauthorized abuses.

In chapter 13, Tad Gonsalves and Kiyoshi Itoh address the operational costs of service systems. The authors use a cost function consisting of service costs and waiting costs, and apply the Particle Swarm Optimization algorithm in order to minimize those final costs. The authors present an example for a practical service system, and reference experimental results to support their findings.

In chapter 14, Roger Y. Lee and Haeng Kon Kim describe an investigation into the application of Model-driven Architecture (MDA) and generative Service-oriented Architecture (SOA) to web services (MS2Web). The authors posit that MDA is able to provide a precise framework for generative web service production. They conclude with a proposed case study that illustrates how the two architectures can be brought together to this end.

In chapter 15, Lihua Liu and Zhengjun Cao present a reanalysis of a traditional verifiable encryption scheme. The authors point out flaws in this scheme and propose solutions for a more secure encryption method.

In chapter 16, Thandar Thein et al. investigate the phenomenon of software aging, whereby software applications with continuous execution for a long period of time display degraded performance and/or an increased rate of hang/crash failures. The authors present a model to evaluate the effectiveness of proactive fault management approached aimed at preventing software aging, and make use of virtualization technology to express downtime and the downtime associated with them.

In chapter 17, Smriti Agrawal et al. present a general scheduling algorithm which offers less energy consumption for weakly hard real-time systems. The authors propose a two phase plan for energy use minimization and present the results of a simulation that demonstrates the effectiveness of the approach.

In chapter 18, Naohiro Ishii et al. propose a modification to the k-nearest neighbor (k-NN) classifier. The authors combine the classifier with both relearning and ensemble methods to provide higher generalization accuracy than conventional algorithms.

In chapter 19, Kazunori Iwata et al. present a method for evaluating agent cooperation in a Multi-Agent System. The authors bring us this analysis via testing using the RoboCupRescue simulation, and present and analyze the final results.

In chapter 20, B. Chandra Mohan et al. discuss the increasing problem of congestion in today's computer networks. The authors propose a prediction model-based data mining approach to finding reliable, congestion-free paths in these networks.

In Chapter 21, Gongzhu Hu provides an analysis of formal specification in UML. The author suggests that a lack of unified component definitions leads to imprecise interpretations. He presents a formal specification for class and state diagrams as a solution to this problem.

It is our sincere hope that this volume provides stimulation and inspiration, and that it be used as a foundation for works yet to come.

May 2008 Roger Lee

Contents

List of Contributors

Smriti Agrawal
Motilal Nehru National Institute of Technology, India
agrawal.smriti@gmail.com

Jinyoung Ahn
Hughes Network Systems, USA
jinyoung123@gmail.com

Yongguang Bao
Aichi Information System, Japan
baoyg_860@hotmail.com

Ilham Berrada
Mohammed V University Souissi II, Morocco
iberrada@ensias.ma

Julien Bourgeois
University of Franche-Comté, France
julien.bourgeois@univfcomte.fr

Zhengjun Cao
Shanghai University, China
caozhj@yahoo.cn

Juno Chang
Sangmyung University, Korea
jchang@smu.ac.kr

Sung-Do Chi
Korea Aerospace University, Korea
sdchi@kau.ac.kr

Belal Chowdhury
La Trobe University, Australia
bchowdhury@psolution.com.au

Morshed U. Chowdhury
Deakin University, Australia
muc@deakin.edu.au

Junchul Chun
Kyonggi University, Korea
jcchun@kgu.ac.kr

Clare D'Souza
La Trobe University, Australia
C.Dsouza@latrobe.edu.au

Rachid Elmeziane
Mohammed V University Souissi II, Morocco
meziane@ensias.ma

Tad Gonsalves
Sophia University, Japan
t-gonsal@sophia.ac.jp

Gongzhu Hu
Central Michigan University, USA
hu1g@cmich.edu

Naohiro Ishii
Aichi Institute of Technology, Japan
ishii@aitech.ac.jp

Valérie Issarny
INRIA, France
valerie.issarny@inria.fr

Nobuhiro Ito
Aichi Institute of Technology, Japan
n-ito@aitech.ac.jp

Kiyoshi Itoh
Sophia University, Japan
itohkiyo@sophia.ac.jp

Kazunori Iwata
Aichi University, Japan
kazunori@vega.aichiu.ac.jp

Aman Jantan
Universiti Sains Malaysia, Malaysia
aman@cs.usm.my

Mahmoud Jazzar
Universiti Sains Malaysia, Malaysia
mahmoudj@cs.usm.my

Ismail Kassou
Mohammed V University Souissi II, Morocco
kassou@ensias.ma

Haeng Kon Kim
Catholic University of Daegu, Korea
hangkon@cu.ac.kr

Jin Kim
Hallym University, Korea
jinkim@hallym.ac.kr

Yanggon Kim
Towson University, USA
ykim@towson.edu

Young Woong Ko
Hallym University, Korea
yuko@hallym.ac.kr

Montri Lawkobkit
Dhurakij Pundit University, Thailand
montrilaw@gmail.com

Roger Y. Lee
Central Michigan University, USA
lee1ry@cmich.edu

Sinjae Lee
Towson University, USA
slee5@towson.edu

Wan yeon Lee
Hallym University, Korea
wanlee@hallym.ac.kr

Lihua Liu
Shanghai Maritime University, China
lhliu@yahoo.cn

B. Chandra Mohan
Anna University, India
abc@cs.annauniv.edu

Takéhiko Nakama
Johns Hopkins University, USA
nakama@jhu.edu

Jong Sou Park
Korea Aerospace University, Korea
jspark@kau.ac.kr

Ranvijay
Motilal Nehru National Institute of Technology, India
ranvijay.mnnit@gmail.com

Chotirat Ann Ratanamahatana
Chulalongkorn University, Thailand
ann@cp.eng.chula.ac.th

Francoise Sailhan
University of Franche-Comté, France
sailhan@ieee.org

R. Sandeep
Anna University, India
abc@cs.annauniv.edu

Rama Shankar Yadav
Motilal Nehru National Institute of Technology, India
rsy@mnnit.ac.in

Gihan Shin
Kyonggi University, Korea
dalgundal@naver.com

Vijay Shyamasundar
Towson University, USA
vijayshyamasundar@gmail.com

D. Sridharan
Anna University, India
abc@cs.annauniv.edu

Yeong-Tae Song
Towson University, USA
ysong@towson.edu

Ho Min Sung
Hallym University, Korea
chorogui@hallym.ac.kr

Thandar Thein
Korea Aerospace University, Korea
thandar@kau.ac.kr

Kuniyoshi Toda
Nagoya Institute of Technology, Japan
agentstaff@phaser.elcom.nitech.ac.jp

Dechawut Wanichsan
Chulalongkorn University, Thailand
g49dwn@cp.eng.chula.ac.th

Takahiro Yamada
Aichi Institute of Technology, Japan
v06723vv@aitech.ac.jp

Shaojian Zhu
Towson University, USA
szhu1@towson.edu

Stopping Criterion Selection for Efficient Semi-supervised Time Series Classification

Chotirat Ann Ratanamahatana and Dechawut Wanichsan

Department of Computer Engineering, Chulalongkorn University
Phayathai Rd., Pathumwan, Bangkok 10330 Thailand
{ann,g49dwn}@cp.eng.chula.ac.th

Summary. High-quality classifiers generally require significant amount of labeled data. However, in many real-life applications and domains, labeled positive training data are difficult to obtain, while unlabeled data are largely available. To resolve the problem, many researchers have proposed semi-supervised learning methods that can build good classifiers by using only handful of labeled data. However, the main problem of the previous approaches for time series domains is the difficulty in selecting an optimal stopping criterion. This work therefore proposes a novel stopping criterion for semi-supervised time series classification, together with an integration of Dynamic Time Warping distance measure to improve the data selection during a self training. The experimental results show that this method can build a better classifier that achieves higher classification accuracy than the previous approach. In addition, the extended proposed work is shown to have satisfactory result for multi-cluster and multi-class semi-supervised time series classifier.

1 Introduction

Time series data are ubiquitous, touching almost every aspect of human life, in domains from business, medical, scientific, management, etc. Some traditional application domains include finance (stock market data, credit card usage data), political ratings, weather data, medical data (electrocardiogram, blood pressure, gait data), internet traffic data, science (aerospace launch telemetry, satellite sensor data), product sales records, etc. It has been recently shown that multimedia data can also be transformed into one or two dimensional time series data [9].

This paper focuses on time series classification, one of the most fundamental tasks in Data Mining. Classification maps unseen input data into predefined groups of the training data. It is often referred to as supervised learning, as the classes (labels) are determined a priori; a set of these predefined data is used in a training process, and learning is made to recognize patterns of interest. However, in many real-life applications and domains, labeled training data, especially positive data, are difficult to obtain, while unlabeled data are largely available. One classic example is in a medical domain, where classification of heartbeats into normal and abnormal signals is typical. Apparently, positive samples, i.e. abnormal heartbeats, are not very easy to obtain, while negative

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 1–14, 2008.
springerlink.com

samples, i.e. normal heartbeats, are abundant, producing unbalanced distribution of the training data. Classification of abnormal heartbeats in this case may be inaccurate if only a few positive examples are available for training.

While supervised learning is highly effective for classification with sufficient amount of training data, other approaches are required for accurate classification when the number of labeled data is insufficiently small. Several semi-supervised learning methods [4, 16] have been proposed in an attempt to resolve this problem. This approach builds a classifier from labeled and unlabeled data together in the learning process. If a suitable learning method is appropriately selected according to the type of data and domains, semi-supervised learning will outperform the supervised learning and give a more accurate classifier [4, 5, 16].

The strength of the semi-supervised learning is the use of only handful amount of labeled training data. Though several semi-supervised approaches have been proposed, only a few could be used for time series data [7, 14] due to its special characteristic within. In particular, Wei et al. [14] have recently proposed a self-training approach (subset of semi-supervised training) for time series data using Euclidean distance as a similarity measure. This self-training method will train itself by trying to expand the set of labeled data with the most similar unlabeled data while exploiting a good stopping criterion. However, occasionally, the data is added inaccurately or the stopping criterion is imprecise, resulting in a poor-quality classifier.

This work proposes a novel stopping criterion for semi-supervised time series classification, and Dynamic Time Warping distance measure is used to improve the data selection during the labeling process. This approach is shown to give a more accurate classifier than the existing methods.

The rest of this paper is organized as follows. Section 2 reviews some background and related work. Section 3 gives details of the proposed method, followed by a set of experiments in Section 4. Section 5 concludes the work and provides suggestions for future work.

2 Background

As mentioned earlier, several semi-supervised learning approaches have been proposed-Generative Model [8], Graph Based Method [3], Density Based Method [1], Co-Training approach [2], and Self Training approach [7, 12, 15]. This paper will particularly focus on the Self Training approach, as its distance-based similarity measure best suits the special characteristic of time series data.

For time series classification, a similarity measurement between two time series is an essential subroutine. The perhaps most popular distance measure is the Euclidean distance metric because it is fast and easy to implement. Although the Euclidean distance metric is widely known to be very sensitive to distortion in a time axis, this distance metric or its minor variation has been applied to the vast majority of research. The ubiquity of Euclidean distance in the face of increasing evidence of its poor accuracy (for classification/clustering) is almost

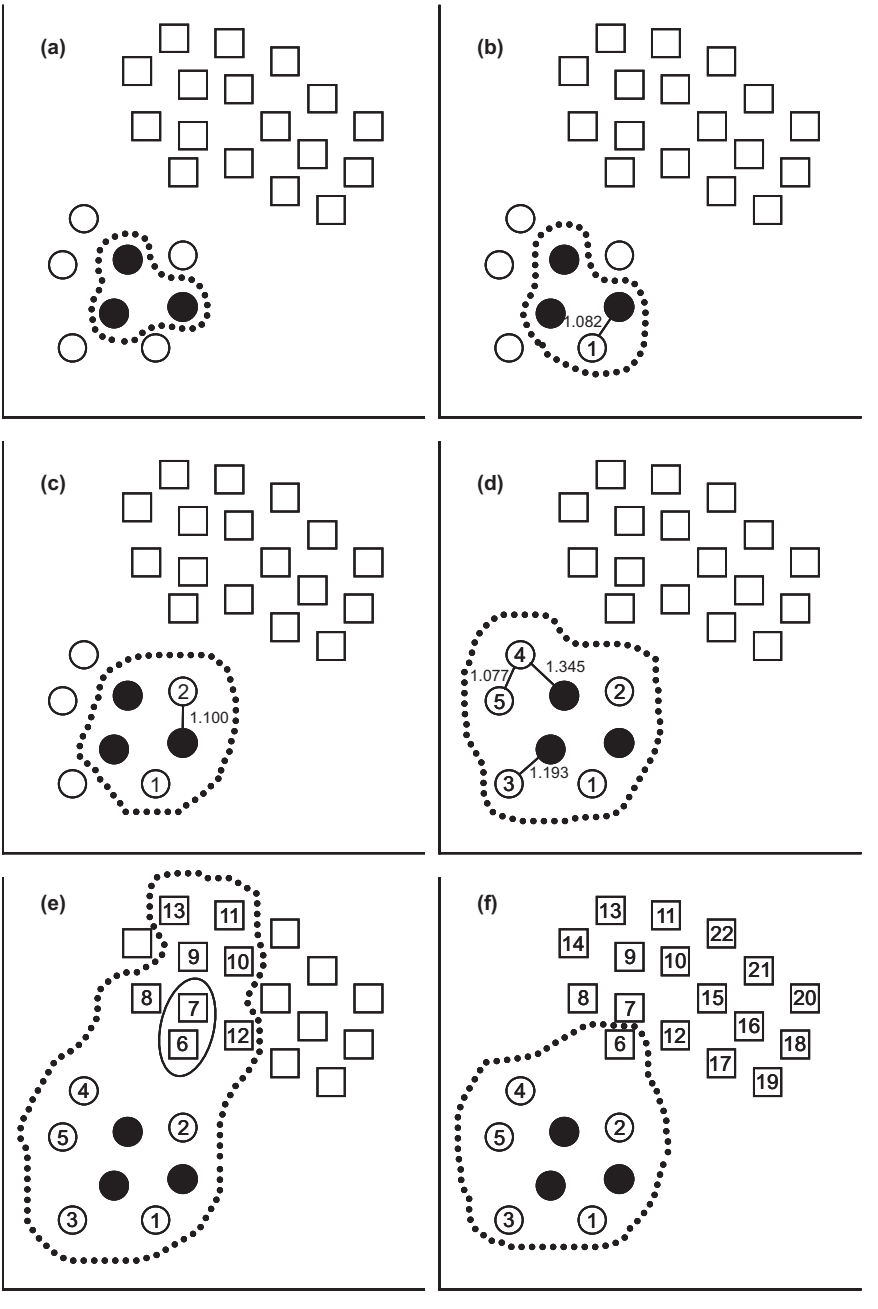


Fig. 1. A stopping criterion based on minimum-distance self-training approach proposed by Wei et al. [14]

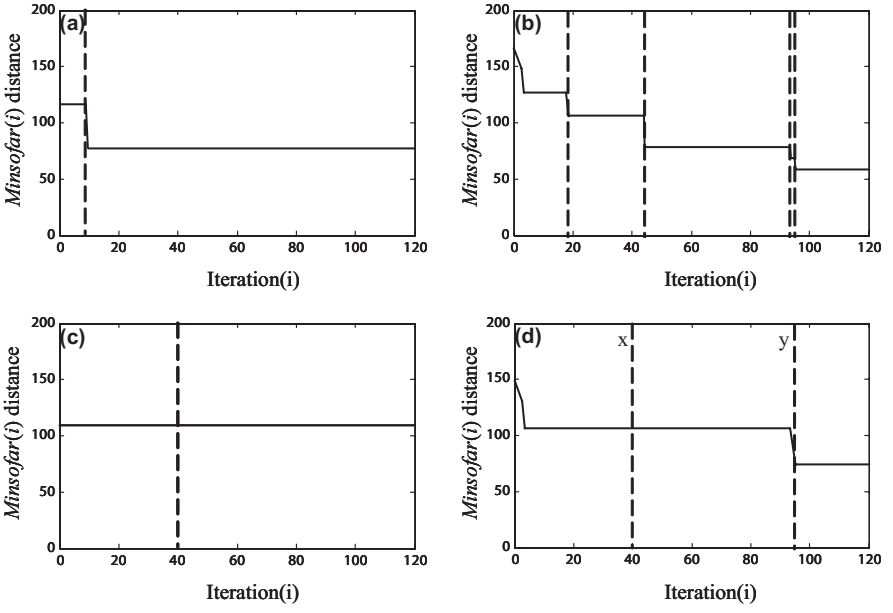


Fig. 2. Problems with current stopping criterion. (**a**) A single drop in the distance in a simple scenario; (**b**) Too many potential stopping criteria; (**c**) No stopping criteria found (should have been at i = 40); (**d**) The stopping criterion occurs too late at y instead of at x.

certainly due to its ease of implementation and the fact that it can readily be indexed with virtually any multidimensional index structure.

The recently proposed self-training approach for time series data [14] also uses the minimum Euclidean distance to build a classifier, as illustrated in Fig. 1. Circles represent positive data (which may be unlabeled), filled circles represent initial positive labeled data, and squares represent negative data. In this example, the 3 labeled positive data is given as an initial positive labeled set in Fig. 1 (a). After the first iteration, the closest data to any member of the positive set is selected in Fig. 1 (b) with their distance recorded. The subsequent iterations are similar, as shown in Fig. 1 (c), (d), and (e). Note that the *Minsofar*(i) is recorded as the minimum distance from the first iteration up to iteration i, as *Minsofar*(1) to *Minsofar*(4) is 1.082, *Minsofar*(5) is 1.077, and so on. The learning continues until every item is moved to the labeled set. At the end, the classifier will select all the data only from the first iteration up to the iteration before the stopping criterion. Their work chooses the stopping criterion at the iteration when overall *Minsofar* occurs the earliest (the first significant drop in *Minsofar*, as shown in Fig. 2 (a). In this example, the first minimum *Minsofar* here happens at iteration 7 (Fig. 1 (e), therefore obtaining the resulting classifier in Fig. 1 (f)).

While this proposed stopping criterion can successfully select the stopping criterion in simple cases (Fig. 1 (a)), it may fail in many others, such as several drops in *Minsofar* distance, constant *Minsofar*, or *Minsofar* occurs too late (optimal stopping criterion is missed), as shown in Fig. 2 (b), (c), and (d), respectively.

3 Proposed Method

Due to several limitations of the existing method, this work aims to improve the stopping criterion selection, and at the same time improve the accuracy of the classifier. One problem with time series data is the typical distortion in the time axis, making Euclidean distance metric unsuitable. However, this problem can be effectively addressed by Dynamic Time Warping (DTW), a distance measure that has long been known to the speech processing community. This method allows non-linear alignments between the two time series to accommodate sequences that are similar but out of phase. In more detail, suppose we have two time series data, a sequence Q and a sequence C which have length m and length n, respectively, where

$$Q = q_1, q_2, \ldots, q_i, \ldots, q_m \tag{1}$$

$$C = c_1, c_2, \ldots, c_j, \ldots, c_n \tag{2}$$

DTW is a dynamic programming technique which calculates all possible warping paths between two time series data for finding minimum distance. Firstly, we create an m by n matrix where every element in matrix is cumulative distance which is defined as

$$\gamma(i,j) = \min d(i,j) + \{\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1)\} \tag{3}$$

where $\gamma(i, j)$ is (i, j) element of matrix that is a summation between $d(i, j) = (q_i - c_j)^2$, a square distance of q_i and c_j, and the minimum cumulative distance of three adjacent element to (i, j).

Next, we choose the optimal warping path which has minimum cumulative distance defined as

$$DTW(Q, C) = \min_{\forall w \in P} \sqrt{\sum_{k=1}^{K} W_k} \tag{4}$$

where P is a set of all possible warping paths, w_k is (i, j) at k^{th} element of the warping path, and K is the length of the warping path. Additionally, a global constraint to DTW is required for a more accurate distance measure, a well-known Sakoe-Chiba band [11], as shown in Fig. 3, is incorporated into our proposed work. Due to space limitations, interest readers may consult [9, 10, 11] for more detail about DTW.

Due to evident advantages of DTW for time series data, our proposed work does incorporate DTW distance measure into our algorithm, as described in Fig. 4.

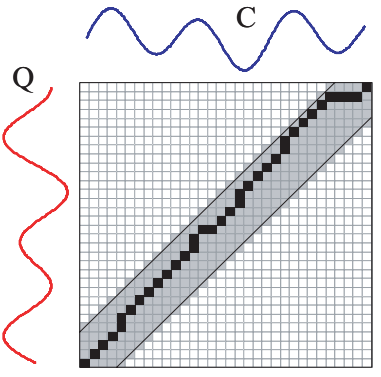


Fig. 3. DTW with Sakoe-Chiba global constraint

Algorithm: Self-training(trainingdata)

1. Split training data into a labeled set and an unlabeled set.
2. Select the time series data from the unlabeled set that has the minimum Dynamic Time Warping distance to any one of the data in the labeled set.
3. Move the selected data from step 2 into the labeled set. Record necessary information for stopping criterion selection.
4. Check number of items in the unlabeled set
   a. If the set if not empty, repeat steps 2-4.
   b. If the set if empty, move to step 5.
5. Determine an optimal stopping criterion.
6. Build a classifier based on the stopping criterion from step 5.

Fig. 4. Self-training approach for 2-class classifier

The first step is to identify the labeled positive data, which would be used as an initial positive training data in the classifier. This could be as few as initial instance (as conducted in our experiment). After that, we try to bring in the best potential data that should subsequently become part of the positive data based on the Dynamic Time Warping distance between the candidate data from the unlabeled set to any of the positive data in the labeled set. Once selected, this distance is recorded. We repeat all the steps until all items in the unlabeled data are exhausted. The distance information from each round is then

used to choose an optimal stopping criterion. Specifically, we propose a Stopping Criterion Confidence (SCC) value as follows:

$$SCC(i) = \frac{|Mindist(i) - Mindist(i-1)|}{Std(Mindist(1)\ldots Mindist(i))} \times \frac{NumInitialUnlabeled - (i-1)}{NumInitialUnlabeled} \quad (5)$$

where $MinDist(i)$ is the minimum distance from step 2) in iteration i, $Std(X)$ is a standard deviation, and *NumInitialUnlabeled* is the number of unlabeled data at the beginning of the learning phase. These standard deviation and the number of unlabeled items are used as the learning weight; larger weights are assigned to early moved items, and the weight is gradually decreased as the learning progresses. Based on the proposed criterion, the achieved SCC of a WordSpotting dataset is shown in Fig. 5, which discovers an optimal stopping criterion in iteration 151 (i that gives maximum SCC value), whereas the stopping criterion from the existing method [14] would find the stopping criterion too early at iteration 13.

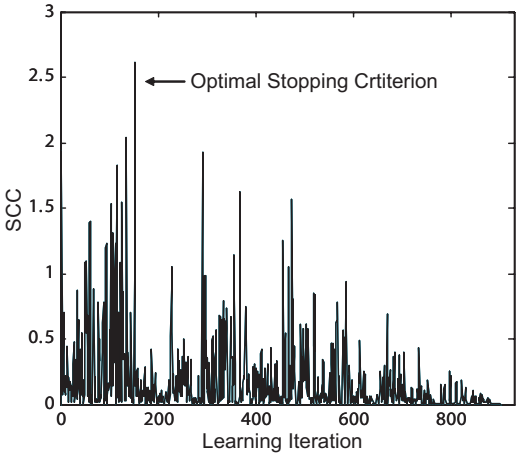


Fig. 5. SCC values from each learning iteration

Once the optimal stopping criterion iteration is determined, the classifier is built by taking all the selected data from iteration 1 up to the optimal iteration i-2. By using SCC instead of *Minsofar* values, our proposed algorithm can eliminate all the problems discussed in Fig. 2 and can be extended to a more complex problem of multi-cluster and multi-class data, as will be demonstrated in Section 4.

The obtained classifier is then used to classify the test data, according to the algorithm in Fig. 6.After the classification, we evaluate the classifier by measuring the precision, recall, and the F-measure of the retrieval. The precision is the ratio between the correctly classified positive test data and the total number of test instances classified as positive. The recall is the ratio between the correctly

Algorithm: Classify(Classifier, testdata)

For each instance of the test data
  a. Calculate the Dynamic Time Warping Distance to every instance in the classifier, and select the instance, *inst* as its one-nearest-neighbor, with minimum distance, *MinDist*.
  b. If *MinDist* ≤ *Threshold*, classify this test data as positive (the same class as *inst*).
  c. Otherwise, classify this test data as negative.
Endfor

Fig. 6. Classifying algorithm

classified positive test data and the total number of all positive instances in the test dataset. However, since each retrieval may yield different precision values at different recall levels, an F-measure is used to give a big picture how different classifiers compare. An F-measure is a ratio between 2*Precision*Recall and Precision+Recall. Generally, the higher the F-measure, the better the classifier.

4 Experimental Evaluation

We evaluate the utility of our proposed method with 10 benchmark datasets on various domains from the UCR Time Series Data Mining archive [6] and the Self-training datasets [13] that are publicly available online. The details of the datasets are shown in Table 1.

Table 1. Details of the datasets used in the experiment

Dataset	Training Data	Test Data	Length	Number of Classes
ECG	810	1216	87	2
Word Spotting	805	905	272	2
Yoga	312	306	428	2
Gun	125	125	152	4
Coffee Cup	28	28	286	2
Olive Oil	30	30	570	4
CBF	465	465	128	3
2-Patterns	1000	4000	128	4
Nuclear Trace	100	100	275	4
Synthetic Ctrl	300	300	80	6

Our main objective is first to build a classifier for 2-class data-positive (class of interest) and negative (everything that is not part of the class of interest). However, the benchmark datasets here are the classification datasets, so that many of them (6 out of 10) have more than two classes. Therefore, for every datasets with more than 2 classes, we choose the data with class label = 1 as our positive examples and the rest as negative examples, following the experimental construction in the previous work by Wei et al. [14].

In the first experiment, we start off the learning phase by randomly select just one instance from the class of interest, and treat it as our only initial labeled data; the rest are treated as unlabeled data for the training. After the learning is complete and the classifier is obtained, it is used to classify the test data and measure the results in terms of precision, recall, and F-measure. We repeat the experiment 30 times for each dataset, so that we get different initial time series data for the learning phase.

With only one positive initial instance for learning, we achieve quite impressive results, as shown in Table 2. Our proposed work has demonstrated to largely outperform the previous approach by a wide margin, in terms of Precision, Recall, and F-measure on every dataset, with one exception in the Coffee Cup dataset. However, the F-measure of this Coffee Cup dataset using previous work's approach itself is not very high; this does reflect the fact that the nature or characteristic of the dataset might not be suitable for this particular semi-supervised classification approach.

Table 2. Experiment results for a single-cluster problem, comparing our proposed method to the previous work (using one initial instance)

Dataset	Previous work			Our Proposed Method		
	Precision	Recall	F-measure	Precision	Recall	F-measure
ECG	0.7284	0.2805	0.4051	0.9690	0.7406	**0.8395**
Word Spotting	1.0000	0.1621	0.2789	0.8847	0.4972	**0.6367**
Yoga	0.8027	0.3285	0.4662	0.8983	0.4808	**0.6263**
Gun	1.0000	0.3636	0.5333	1.0000	0.7333	**0.8462**
Coffee Cup	0.5551	0.4451	**0.4940**	0.6566	0.2238	0.3338
Olive Oil	0.6846	0.4000	0.5050	0.8667	0.5056	**0.6386**
CBF	0.9926	0.1118	0.2010	1.0000	0.1824	**0.3085**
2-Patterns	0.7738	0.0887	0.1592	1.0000	0.6283	**0.7717**
Nuclear Trace	0.5027	0.2917	0.3691	1.0000	1.0000	**1.0000**
Synthetic Ctrl	1.0000	0.2707	0.4260	1.0000	0.8200	**0.9011**

It is also worth noting that our proposed method also works when there exist several clusters within the data, as indicated in the experiment's setting that the negative set may contain several classes of data, as illustrated in Fig. 7. Particularly, in the case where there are more than one clusters in the positive class, if at least one initial labeled learning instance comes from every cluster, the resulting classifier turns out to be quite accurate.

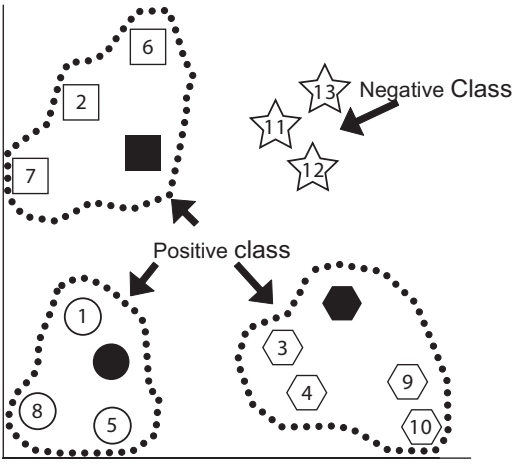


Fig. 7. A multi-cluster example where several clusters exist within the positive class

Table 3. Experiment results for a multi-cluster problem, comparing our proposed method to the previous work (using five initial instances)

Dataset	Previous work			Our Proposed Method		
	Precision	Recall	F-measure	Precision	Recall	F-measure
ECG	0.3158	0.8451	0.4598	0.9667	0.7447	**0.8601**
Word Spotting	0.9258	0.4832	0.6350	0.9841	0.5278	**0.6871**
Yoga	0.8048	0.4573	0.5832	0.6919	0.9107	**0.7863**
Gun	0.9661	0.5933	0.7352	1.0000	0.7333	**0.8462**
Coffee Cup	0.6220	0.7120	**0.6643**	0.6092	0.6671	0.6308
Olive Oil	0.5487	0.7778	0.6435	0.9408	0.8222	**0.8775**
CBF	0.9401	0.3886	0.5499	0.8402	0.6684	**0.7445**
2-Patterns	0.7391	0.2493	0.3729	1.0000	0.6907	**0.8170**
Nuclear Trace	0.4614	0.4000	0.4285	1.0000	1.0000	**1.0000**
Synthetic Ctrl	0.6617	0.3195	0.4309	1.0000	0.8200	**0.9011**

Fig. 7 depicts an example of a scenario where several clusters may exist within a positive class, shown as circles, squares, and hexagons. The filled objects represent initial labeled data for training. As mentioned earlier, if these initial data come from each and every positive-class cluster, we can achieve a more accurate classifier. In this particular example, three labeled positive data from each positive-class cluster are assigned as an initial labeled set. The rest of the process is quite similar to that or our original proposed algorithm; unlabeled data are gradually selected accordingly and moved into the labeled set. The numbers specified within the objects denote the orders whose items are moved into the labeled set. The desired stopping criterion in this particular example is expected to be at iteration 11, as it correctly splits positive- and negative-class examples.

Table 4. Experiment results for a multi-cluster problem, comparing our proposed method to the previous work (using ten initial instances)

Dataset	Previous work			Our Proposed Method		
	Precision	Recall	F-measure	Precision	Recall	F-measure
ECG	0.4616	0.7736	0.5782	0.9668	0.8106	**0.8818**
Word Spotting	0.8480	0.7486	0.7952	0.9837	0.7713	**0.8646**
Yoga	0.6346	0.9744	0.7686	0.7326	0.9081	**0.8110**
Gun	1.0000	0.6578	0.7936	1.0000	0.7333	**0.8462**
Coffee Cup	0.6549	0.6821	**0.6682**	0.6092	0.7333	0.6655
Olive Oil	0.6444	0.8167	0.7204	0.9408	0.8556	**0.8962**
CBF	0.7966	0.6843	0.7362	0.9848	0.8426	**0.9082**
2-Patterns	0.6617	0.3195	0.4309	1.0000	0.7539	**0.8597**
Nuclear Trace	0.4641	0.6583	0.5444	1.0000	1.0000	**1.0000**
Synthetic Ctrl	0.6803	0.9267	0.7846	1.0000	0.8200	**0.9011**

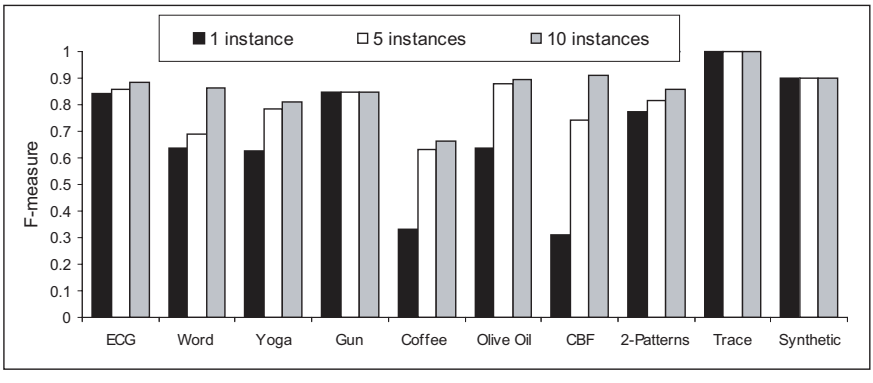


Fig. 8. The effect of the number of initial training instances on quality of the classifier

We also repeat the experiment with 5 and 10 instances of initial data to see if this would have any effects on the quality of the classifier, whose results are shown in Table 3 and Table 4, respectively. Both reveal that higher number of initial data would result in a more accurate classifier, as shown in Fig. 8.

Table 3 and Table 4 show the precision, recall, and F-measure of each dataset, comparing the previous work on self-training approach with our proposed method, averaged over 30 runs, demonstrating that our proposed method also outperform the previous work by a wide margin.

4.1 Multi-class Self-training Classifier

We extend the experiment further to build a multi-class classifier where the positive data or data of interest may consist of data from many classes, and we would like to also tell them apart, not just knowing that they are positive. This

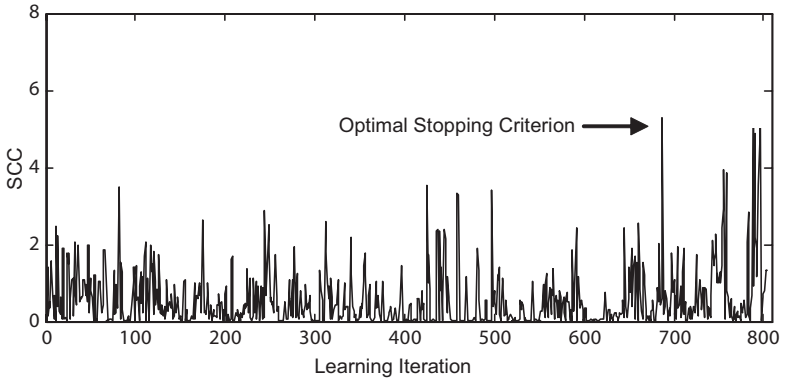


Fig. 9. SCC values from each learning iteration of the proposed multi-class self-training approach

Table 5. Experiment results comparing the multi-class proposed method with the supervised training

Dataset	Accuracy(%)		
	Supervised Learning	Multi-class Self-training proposed method (best case)	Multi-class Self-training proposed method (average case)
ECG	99.01	79.76	79.37
Word Spotting	100.00	83.24	80.66
Yoga	100.00	82.60	80.68
Gun	88.80	**90.40**	86.08
Coffee Cup	86.77	**90.00**	82.33
Olive Oil	82.01	67.85	63.21
CBF	99.07	66.02	63.65
2-Patterns	100.00	99.70	91.79
Nuclear Trace	100.00	87.00	84.40
Synthetic Ctrl	99.03	91.00	87.46

multi-class self-training approach works very similarly to the original approach. However, typical self-training approaches are designed for a 2-class problem, where the classifier will return either positive or negative as the answer. In our extension to a multi-class classifier, this can be used for n-class problems, where the classifier will provide the class label as the answer. Therefore, our weighting term involving the *NumInitialUnlabeled* in the Stopping Criterion Confidence values is removed. In addition, the classification phase is very similar to the algorithm presented in Fig. 6, except that the threshold value in steps b) and c) is no longer needed. Instead, the test data will be classified to have the same class label as inst that gives the minimum one-nearest-neighbor distance. As mentioned earlier, the initial labeled data does play an important role in the quality of this multi-class classifier as well.

The plot of SCC values for ECG dataset is shown in Fig. 9, which discovers an optimal stopping criterion at iteration 678 (i that gives maximum SCC value). Therefore all instances that are moved prior to iteration 678 is considered positive, and the rest is considered negative.

To evaluate the performance, instead of the precision and recall, the accuracy is measured as a ratio between correctly classified instances and the number of all instances in the test data. We compare our results with the ideal case of fully supervised learning approach, as illustrated in Table 5. The results demonstrate that the proposed multi-class approach performs comparably to the supervised learning, and in some cases even outperforms the supervised learning.

5 Conclusion

This work proposes a semi-supervised learning using a self-training approach that integrates together a good stopping criterion selection and the Dynamic Time Warping distance measure. The results validate the utility of our approach, as it yields a better classifier that achieves higher classification accuracy than the existing work. Additionally, this work is shown to be effective for both multi-cluster and multi-class classifier. As a future work, to make learning process faster, early abandonment can be adopted such that the number of learning iteration is reduced and only fraction of the training data is needed for the classifier.

References

1. Bennett, K.P., Demiriz, A.: Semi-Supervised Support Vector Machines. In: Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II (1999)
2. Blum, A., Lafferty, J.: Learning from Labeled and Unlabeled Data using Graph Mincuts. In: Proceedings of 18th International Conference on Machine Learning (2001)
3. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In: Proceedings of 11th Annual Conference on Computational Learning Theory, Madison, Wisconsin, United States (1998)
4. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
5. Cohen, I., Cozman, F.G., Sebe, N., Cirelo, M.C., Huang, T.S.: Semi-Supervised Learning of Classifiers: Theory, Algorithms, and Their Application to Human-Computer Interaction. IEEE Transaction on Pattern Analysis and Machine Intelligence (2004)
6. Keogh, E.: The UCR Time Series Classification/Clustering Homepage (January 2008), http://www.cs.ucr.edu/~eamonn/timeseriesdata/
7. Li, M., Zhou, Z.H.: SETRED: Self-Training with Editing. In: Proceedings of 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005) (2005)
8. Nigam, K., Mccallum, A.K., Thrun, S., Mitchell, T.: Machine Learning (2000)

9. Ratanamahatana, C.A., Keogh, E.: Everything you know about Dynamic Time Warping is wrong. In: Proceedings of 3rd Workshop on Mining Temporal and Sequential Data, In Conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004) (2004)
10. Ratanamahatana, C.A., Keogh, E.: Making Time-Series Classification More Accurate Using Learned Constraints. In: Proceedings of SIAM International Conference on Data Mining (2004)
11. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. Morgan Kaufmann, San Francisco (1990)
12. Shahshahani, B.M., Landgrebe, D.A.: The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon. IEEE Transactions on Geoscience and Remote Sensing (1994)
13. Wei, L.: Self Training dataset (May 2007), `http://www.cs.ucr.edu/~wli/selfTraining/`
14. Wei, L., Keogh, E.: Semi-Supervised Time Series Classification. In: Proceedings 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
15. Zhang, R., Alexander, I.R.: A New Data Selection Principle for Semi-Supervised Incremental Learning. In: Proceedings of 18th International Conference on Pattern Recognition (ICPR 2006) (2006)
16. Zhu, X.: Semi-Supervised Learning Literature Survey. Technical report, no.1530, Computer Sciences, University of Wisconsin-Madison (2005)

Enhancing the Blockage of Spam over Internet Telephony (SPIT) Using Adaptive PMG Algorithm

Jinyoung Ahn, Vijay Shyamasundar, and Yeong-Tae Song

Hughes Network Systems, Towson University
100 Lakeforest Boulevard Gaithersburg, MD 20877
jinyoung123@gmail.com

Summary. Voice over IP (VoIP) is undoubtedly one of the most rapidly emerging technologies in today's telecommunication industry. With its widespread acceptance, VoIP is all set to redefine the way world communicates. However, there is a huge potential threat from spammers who could take advantage of its wide spread acceptance and cost effectiveness of VoIP. Even though SPam over Internet Telephony (SPIT) is still considered as an emerging problem, it is gaining increased attention from several research communities, universities and industries. Various algorithms and techniques have been proposed to combat SPIT. In this paper we focus on the Progressive Multi Gray level (PMG) approach proposed by Shin et al [1]. We have analyzed the behavior of the PMG algorithm for various call patterns and proposed a way of fine tuning the performance of the PMG algorithm for better blockage of SPIT.

1 Introduction

VoIP is termed as the most disruptive technology in today's telecommunication industry as it has the potential to alter the technical and economic dynamics of the market. The cost effectiveness and ability to integrate other advanced features like video conferencing are the major driving factors for the widespread acceptance of VoIP.

Realizing this wide spread acceptance of VoIP, the spammers are gaining ground to use this as a platform for next generation spam. VoIP is more vulnerable for spamming than PSTN. The caller's identity cannot be retraced easily in VoIP like in PSTN. PSTN spamming is also more expensive for spammers. VoIP unlike email is time critical and does not tolerate any latency. It is difficult to introduce any sophisticated spam blocking techniques to scrutinize the call before it is terminated. Hence SPIT is considered to be more threatening than its email counterpart. The fact that the voice is more susceptible to spamming than email is being exploited by the today's spammers. Even though SPIT is still considered as emerging problem several SPIT incidents are already being reported in Japan which is a more mature VoIP market than North America today [3]. A major VoIP provider, SoftbankBB found three incidents of SPIT within its network, which included unsolicited commercial messages for an adult Web site and illegitimate requests for personal information. NEC is predicting 40% to 70% of the phone calls would be SPIT soon [6]. Even though some other giants in the telecommunication industry might seem skeptical about these numbers predicted by NEC, most of them seem to believe in the rapid rise of SPIT in the near future.

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 15–26, 2008.
springerlink.com

In this paper, we discuss the behavior of the original PMG algorithm for several call patterns and highlight its limitation. We have proposed a solution to overcome the limitations of the PMG algorithm to block SPIT more efficiently. We also demonstrate our proposed adaptive approach is more efficient than the original PMG algorithm.

2 Research Objective

Shin et al. [1] demonstrated in their experiments "how" to choose the configuration parameter values and to apply to the PMG algorithm. However their paper does not mention "what" values to choose for the configuration parameters like short term period, long term period, constants C1 and C2 to apply to the algorithm in the real world scenario.

Our initial objective was to determine the best set of configuration parameters for PMG algorithm to block SPIT efficiently. We conducted several experiments to test the performance of PMG algorithm using various call patterns and certain configuration parameters. We found out during the course of our experiments that one set of configuration parameters that best fits for blocking spam may fail to perform as intended, for it may also block the calls from a normal user.

Knowing such limitations of the PMG algorithm, we modified our objective to determine a solution to fine tune the performance of the PMG for efficient blockage of SPIT.

3 Related Works

Several researchers and corporate companies have come up with different techniques to combat SPIT. They agreed that SPIT cannot be completely blocked by using one technique alone and the techniques used to block e-mail spam would be of not much help since calls have to be filtered in real time.

NEC Corporation has announced the development of a new SPIT prevention technique called "VoIP SEAL" [8]. They perform the Turing tests to determine if the call is from a human user or a machine and follow a modular approach to develop new modules to tackle new kind of SPIT attacks. They claim to detect and block 99% of SPIT calls with their approach.

Microsoft's approach is based on a voice-recognition technology. According to Microsoft's Horvitz V-Priorities is believed to have 3 levels of filtering [4]. The first level of filtering is done based on the caller's voice characteristics. At the second level, the filtering is done by smart rudimentary word and phrase recognition. Metadata information such as the time and length of a message is used at the third level.

The Eyeball Networks follow a "volume based model" in their AntiSPIT Technology [5]. Eyeball AntiSPIT Server limits the number of calls a caller can make or receive. It also considers the previous caller-callee relationships and calling patterns as a combined technique to block SPIT.

Iotum's approach to block SPIT is based on black or white list. The Iotum Relevance Engine compares the incoming caller's identity to a list of known callers, and simply blocks unknown callers or redirects them to the voice mail [7].

Voice Spam Detection is an approach proposed by Ram Dantu and Prakash Kolan [2]. It is primarily based on Bayesian learning and user feedback. In this approach there is a human element involved in computing the trust to allow or block a caller.

Several other companies and researchers have suggested various methods like universal blacklisting, source filtering, content filtering, challenge-response, and volume based models etc to combat SPIT.

Progressive Gray Level algorithm [1] is a simple yet effective approach to block SPIT without involving any sophisticated voice recognition techniques or human interference for feedback or challenge-response mechanism. This algorithm residing only on the server side seems to be a promising solution to combat SPIT when used with strong user authentication techniques. The PMG algorithm could also be easily integrated with any standard SIP server.

4 The PMG Algorithm

The PMG algorithm is a type of source filtering based on the SIP URI. If a user is not listed as white or black then the algorithm computes a gray level for each user. If the computed gray level for the user is less than the gray level threshold, the call is connected; else the user is temporarily blocked until his gray level falls below the threshold. The beauty of this simple algorithm is it works efficiently by analyzing previous call patterns without any sophisticated voice detection techniques or human interference.

PMG computes short and long term gray levels for each user whenever an incoming call request is received. The call is blocked if the sum of short and long term gray levels exceeds the gray level threshold. The idea behind the short term is to quickly cut down the users generating burst calls in a short period of time to attack the server and the long term is used to counter the repeated periodic burst attacks. The short term gray level rapidly increases with the decrease in the call interval. It also decreases rapidly if the user is idle for a longer period of time. The long term gray level increases or decreases at a much slower pace. Once short term gray level hits the threshold it passes its value to long term gray level and resets itself to zero. Hence the short term gray level protects the call server from burst spam calls. The long term gray level takes the value from short term gray level once the short term gray level reaches the threshold and decreases it very slowly to block the spam calls over a long period of time until it falls below the threshold.

PMG requires the administrator to choose the values for the following configuration parameters to compute the gray level for a user: Gray Level Threshold, Short Term Period, Long Term Period, Constant C1 and Constant C2.

5 Fine Tuning the PMG Algorithm

When PMG is implemented in a real world scenario the administrator will choose the following configuration parameters: gray level threshold, short term period, long term period, constants C1 and C2. Depending on the requirements the administrator could decide how stringently the algorithm should perform. Shin et al. [1] summarized in his

paper on how to set the configuration parameters depending on the situation and objective of the VoIP setting. They also mentioned that the configuration parameters could be customized for each individual user or group of users to achieve more efficiency. However, it is not explained how to implement the customizations. It would also be a very cumbersome approach and hard to maintain or set the policies if the administrators need to customize the configuration parameters for each individual users or even a group of users.

We have identified during the course of our experiments that if an administrator chooses only one set of configuration parameters, the PMG does not work efficiently for all users. This concept works well in blocking the software SPIT generators or heavy duty spammers who generate a large volume of calls or allow normal users who place very few calls. But the algorithm fails to perform as intended in the case of a power user or a potential spammer. It might lead to a risk of blocking a legitimate caller over a period of time or could allow calls from a spammer for a long duration of time to cause enough damage before he is blocked.

5.1 PMG Limitation

The original PMG algorithm has the following limitations:

- A set of configuration parameters that works best to allow normal users would not block spammers efficiently. Another set of configuration parameters that block the spammers efficiently involves a risk of blocking the legitimate normal users over a period time.
- PMG requires customizing of the configuration parameters for each individual user or a group. As the user base increases this process becomes cumbersome for the administrators to maintain different policies for different users or group of users

We will not dwell into other limitations of the original PMG such as its inability to tackle fake identity issues etc. We are only concentrating on the improving the efficiency of SPIT blockage using PMG technique. As Shin et al. mentioned PMG should be supported by strong user authentication methods for it to be effective. Also SPIT blockage can be successful only with a combination of several techniques.

5.2 Proposed Solution

We use the “divide and conquer” approach to overcome some of the limitations of original PMG algorithm we stated in section 5.1. From our experiments we have determined the values of configuration parameters that work well for normal user i.e. it will allow all the calls without blocking a legitimate user there by reducing the false positives for this scenario. We have also determined another set of configuration parameters works best to block spammers. The idea of our approach is to determine the instantaneous behavior of the current caller based on certain threshold that could be decided using the service provider’s historical information. If the instantaneous call frequency of the current user is low, a set of configuration parameters that best suites normal users would be applied to the PMG algorithm. If the instantaneous call frequency of current caller

is high, a set of configuration parameters that best suits the spammers would be applied to PMG algorithm. To achieve this, we implemented a plug in to work on top of the PMG algorithm that instantaneously detects the call frequency of the current caller and determines what set of configuration parameters need to be applied for the PMG algorithm.

We introduce two more parameters short term call density and long term call density in order to determine the call frequency of the user. The short term call density is the number of calls placed by a particular user per minute and long term call density is the number of calls placed by a particular user per day. The short term call density timer is reset every minute and it keeps track of number of calls placed by a particular user within a minute. The long term call density timer is reset every day and it keeps track of number of calls placed by a particular user within a day. A certain threshold could be set for both the short term and long term call density parameters. The service providers could look in to their historical information based on the statistical data they have generated in the past. E.g. looking at past n number of year's data they might be able to safely assume that users would not be able to place more than x calls per minute or y calls per day on an average. Based on that information their administrator could choose a short term call density of x and a long term call density of y. We tried to obtain real world data for several carries for these numbers in vain. Determining the short term call density threshold for our experiments was fairly simple. We tried to generate as many VoIP calls possible with in a minute in our lab and we used a stop clock to time it. Once a call was connected, we used to say "this is call number n" (where n = 1,2,3,4...) and disconnect the call. We observed that it was almost practically impossible to generate more than 4 legitimate calls with in a minute. Long term call density threshold was difficult to come up with as we were not able to obtain any related information from a reliable source. We conducted an opinion poll with in the university and interviewed about 100 people of different age groups to determine the long term call density. The average of the numbers we obtained turned out to be 20 calls per day. This approach might seem very subjective and one might argue considering a call centre scenario where approximately 200 calls might be placed with in a day as they might operate 24/7. In such scenario the service providers might have to depend on white listing technique for such known call centre users as they regularly generate such high volume of calls on a daily basis. The numbers will vary if different values are chosen for short term and long term call densities, but we have observed from our extensive research that this concept still holds good to block SPIT more efficiently than original PMG. The administrators can train their call frequency detectors based on their historical information. In our experiments we have chosen the short term call density threshold to be 4 and long term call density threshold to be 20. The frequency detector uses these two parameters to determine if the user placing the current call is a high or low frequency user. In our case if the user places more than 4 calls per minute OR more than 20 calls per day, he will be treated as a high frequency user and a set of configuration parameters that best suits for a spammer would be applied to PMG to compute the gray level for this user at that particular instant. Otherwise the user is treated as a low frequency user and the configuration parameters that best allow the caller would be applied to the PMG to compute the gray level. The beauty of this

approach is it adapts dynamically with the incoming call pattern. If a normal user places more number of calls than usual then he would be treated as spammer at that instant of time but once he reverts back to his usual number of calls he would be treated as a low frequency user and thus PMG reduces his gray level much more quickly. Hence the normal user would never be blocked even over a long period of time. Also for the real spammer this approach applies the high frequency configuration parameters all the time thus blocking them more efficiently before they placed too many calls.

Following this approach not only eliminates the need for the administrators to set policies for each individual users or user groups but also works more efficiently as more appropriate configuration parameters are applied for different types of users.

6 Experiment Methodology

The PMG algorithm residing on any SIP server is triggered when an INVITE message is received. PMG gets the local system time on which it is installed upon receiving the INVITE message and determines the call interval between this call and previous call placed by the same user. It then analyzes the previous call patterns and computes a gray level for that particular user using a set of configuration parameters. PMG passes the INVITE message to the concerned SIP server if the computed gray level for that user is less than the threshold or it will just block the INVITE message from getting to the SIP server.

Since PMG relies only on INVITE message to combat SPIT, it could be easily integrated into any SIP server to block SPIT efficiently. Also Shin et al. mentioned in his paper that the PMG algorithm does not depend on implementation details and was successfully integrated and tested with Cisco Call Manager, IPTEL SER and VOCAL.

Our research mainly concentrates on fine tuning the performance of PMG itself without going into the details of integrating the algorithm with different SIP servers. For the purpose of our experiments it would be very time consuming and tedious to generate real calls to test the performance of PMG. More over, all the PMG algorithm needs is an INVITE message to trigger upon. Hence we have used an input text file with the SIP URI representing a particular user with several time stamps indicating the time at which a call request was placed to the SIP server. A sample input file is as shown below:

```
Uri-bob@towson.edu
2006 7 1 2 32 00
2006 7 1 3 12 00
.....................
```

The first line of the file represents the user's SIP URI. The numbers in each line represents the time at which the INVITE message was received by the PMG algorithm from bob. The time stamps are in the following format:

Year Month Day Hours Minutes Seconds

These time stamps simulate the time that PMG gets from the local system on which it is installed upon receiving a SIP INVITE message from a user. An Input file is generated for several scenarios described below with 1000 call requests for each type of user. The scenarios chosen below are all subjective and our best effort to simulate different real

world scenarios to prove that Adaptive PMG will effectively reduces the false positives that could have occurred in original PMG.

The following call scenarios were simulated using the input files to verify the performance of PMG algorithm:

- **Normal User.** This scenario represents a regular residential user placing any where between 1 and 10 calls per day with an average of about 5 calls per day.
- **Power User 1.** This scenario is used to represent a residential user who places 10 to 20 calls every day with an average of 15 calls per day.
- **Power User 2.** This scenario represents a burst mode of calls from a normal residential user perhaps due to an emergency situation. This is very similar to Power User 1 placing 10 to 20 calls per everyday but 40 to 60 calls were placed in a day for about 4 days due to an emergency situation thus increasing the call density. This emergency situation has been assumed to occur every month just to stress test the PMG and make sure the normal users are never blocked i.e. to reduce the false positives even if that would give an edge to spammers.
- **Spammer.** This scenario represents a human spammer working an 8 hour shift and placing 40 to 60 calls averaging about 48 calls per day.
- **Software SPIT Generator.** This represents a software attack on the server placing on an average of 300 calls everyday.

PMG algorithm is verified for the scenarios mentioned above using several sets of configuration parameters. The idea here is to determine the best set of configuration parameters to block SPIT and allow normal users efficiently. Several sets of configuration parameters including gray level threshold, short term period, long term period, constants C1 and C2 are saved in a text file and applied to the PMG to compute the gray level.

PMG algorithm was implemented using Java. The input text files, one with user call patterns and the other file containing a set of configuration parameters were provided as the input for this java code to compute the gray level of the user and decide whether to block or allow his call. MySQL database was used to maintain the user's history. The java code always refers to the database when there is a call request from the user. It takes into account the previous records for that user while computing the gray level to decide to allow or block the call. It updates the database every time the new values are computed for that user. The java program logs the output to a .CSV file to provide the details of the long term values, short term values, the number of connects and disconnects etc. The data obtained from the output .CSV files were transformed to graphs to analyze the behavior of PMG.

7 Original PMG Results

In this section we discuss the behavior of original PMG algorithm for different sets of configuration parameters. We also determine the best set of configuration parameters that works well for the normal users and spammers.

Our initial objective for the experiments was to determine the best values for short and long term periods (STP and LTP) while keeping C1 and C2 constant. Thus the

behavior of PMG was observed by varying the values of STP and LTP while keeping C1=C2=1 and Gray Level Threshold T = 240000.

It was observed from the results that higher values of STP are desirable to block spammers efficiently in contrast to the lower values of STP that best fits the bill to allow normal users. Higher the value of LTP longer is time that a user has to wait after being blocked for reaching the threshold. Hence the value of LTP does not matter much until the user reaches the threshold once. Our findings from initial experiments have been summarized in the table below:

Table 1. Experiment1 - Configuration Matrix

STP-LTP Scenario	1min 1hr	15m 1hr	30m 2hrs	1hr 3hrs	3hrs 6hrs	6hrs 12h	12hrs 24hrs
Normal User	++	++	++	++			
Power User1	++	++					
Power User2	++	++	+				
Spammer						+	+
S/W SPIT					+	+	++

NOTE: + is desirable, ++ is more desirable.

In the next set of experiments, the objective was to study the impact of constants C1 and C2 on the algorithm. It was observed that the PMG algorithm works more efficiently to block spammers from quick burst attacks with higher values of C1. The table below summarizes the best values of STP and LTP with C1 = 6, C2 = 1 and T = 240000.

Table 2. Experiment2 - Configuration Matrix

STP-LTP Scenario	1min 1hr	15m 1hr	30m 2hrs	1hr 3hrs	3hrs 6hrs	6hrs 12h	12hrs 24hrs
Normal User	++	++	++	++			
Power User1	++	+					
Power User2	++	+					
Spammer					++	++	
S/W SPIT					++	++	+

NOTE: + is desirable, ++ is more desirable.

7.1 Summary of Findings

The need for different set of configuration parameters for spammers and normal users was evident from the result of this experiment. It was observed that a STP of less than 15 minutes with C1 = C2 = 1 would be most optimized for normal users. We also observed that spammers could be blocked more quickly choosing C1 greater than C2 and STP greater than at least 6 hours. Also, choosing a higher value for long term period keeps the blocked spammer from spamming for a long time before reducing his gray level below the threshold.

8 Adaptive PMG Results

8.1 Adaptive PMG Experiment

In this section we discuss the results obtained after applying our proposed plug in to the PMG algorithm i.e. the results of Adaptive PMG. The following configuration parameters have been used for the Frequency Detector:

Short term call density = 4 (number of calls per min)
Long term call density = 20 (number of calls per day)

After carefully observing the results of PMG experiments 1 and 2, the following configuration parameters were chosen in our Adaptive PMG experiments:

- PMG Normal User configuration parameters:
  T=240000, C1 = 1, C2 = 1, STP = 15mins, LTP = 1hr
- PMG Spammer configuration parameter:
  T=240000, C1 = 6, C2 = 1, STP = 6hrs, LTP = 12hrs

Scenario 1 - Normal User

As seen from the graph above, a normal user generating about 1 to 10 calls every day would never be blocked. Even after 1000 calls the gray level for this type of user would not even cross 16 versus the allowable limit of 240000.

Scenario 2 Power User 1

A power user generating about 10 to 20 calls every day would never be blocked either as shown in the Figure 1. Even though the gray level of this type of user increases more rapidly it also decreases quickly with the configuration parameters chosen as shown above.

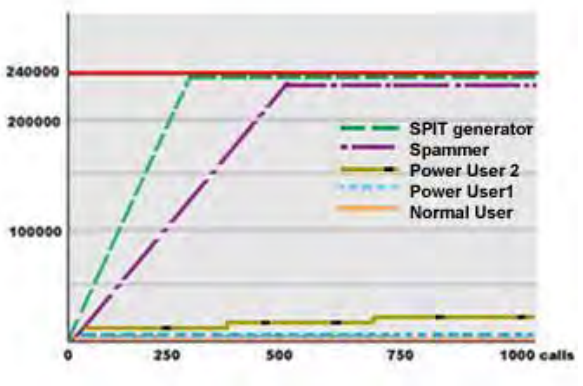


Fig. 1. Adaptive PMG behavior.
Low frequency configuration: c1=1 c2=1, STP = 15mins, LTP=1hr.
High frequency configuration: c1=6 c2=1, STP = 6hrs, LTP=12hrs.

Scenario 3 Power User 2

In this scenario, we stress test the performance of Adaptive PMG by adding burst mode calls for the above power user 1 scenario. Power User 1 and 2 are the scenarios that would not be efficiently handled by the original PMG algorithm alone. Figure 1 confirms that the Adaptive PMG approach could block SPIT more efficiently. Even for the stress test input the maximum gray level reached for this type of user is around 23400 versus the allowable threshold of 240000. If this call pattern is from a legitimate user, it is believed that he reverts back to normalcy after a few days and then he would be treated as a normal user and his gray level would be decreased more drastically. However this type of user could only be blocked if this call pattern persists for around 10 months. It would be practically not possible for a legitimate user to have this emergency situation every month and this would be too little number of calls if this is an attempt by a spammer. However, this performance is desired as we would never want to block a legitimate user even if that would mean to allow some spam calls.

Scenario 4 Spammer

The graph in Figure 1 depicts the performance of Adaptive PMG for a spammer generating about 40 to 60 calls every day between 9 am to 5pm. The Adaptive PMG blocks this user after allowing 584 calls. Thus the Adaptive PMG needs about 10 days to block a user generating about 50 calls per day. This could be stopped much quickly by choosing a much higher value for STP and LTP. However, doing so might lead to a risk of blocking a legitimate user placing more calls than usual due to an emergency situation. Even though the higher values of STP or LTP blocks SPIT more quickly we recommend choosing the most optimized value depending on the situation. If the administrator decides to be more aggressive on spammers, the Adaptive PMG provides him the privilege to choose higher values of STP and LTP for spammer configuration parameters.

Scenario 5 SPIT Generator

This scenario shows the behavior of Adaptive PMG for software generated SPIT attack placing an average of 300 calls per day. In this experiment the software SPIT generator approximately generates about 12 calls per hour. This type of attack would be blocked after placing 250 calls i.e. not even allowing them to spam for more than a day. Usually the software SPIT attacks generate a lot more calls than assumed here. But we are considering testing the worst case scenario to make sure it works. However, if more calls are placed than this the Adaptive PMG would block it much quickly.

8.2 Comparing PMG and Adaptive PMG

In this section, we compare the performance of PMG versus Adaptive PMG and demonstrate how Adaptive PMG resolves the false positives and false negative issues with the original PMG algorithm.

From the above graph you can see that using PMG, calls from a normal user are never going to be blocked. However, you can also see that PMG would not block the

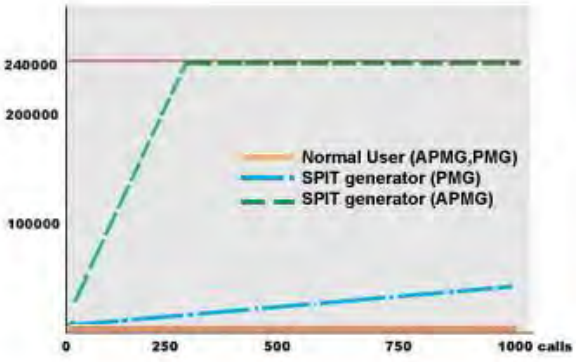


Fig. 2. Adaptive PMG vs PMG.
PMG: c1=1 c2=1, STP = 15mins, LTP=1hr.
APMG Low frequency configuration: c1=1 c2=1, STP = 15mins, LTP=1hr.
APMG High frequency configuration: c1=6 c2=1, STP = 6hrs, LTP=12hrs.

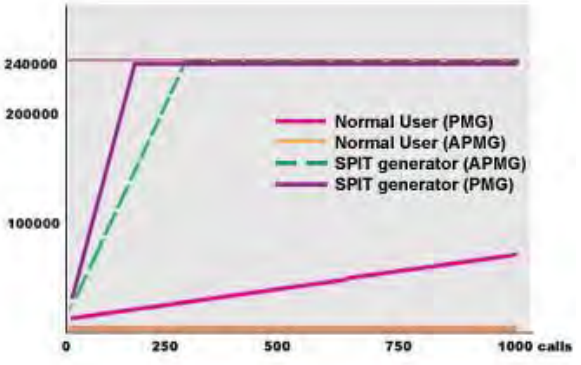


Fig. 3. Adaptive PMG vs PMG.
PMG: c1=6 c2=1, STP = 6hrs, LTP=12hrs.
APMG Low frequency configuration: c1=1 c2=1, STP = 15mins, LTP=1hr.
APMG High frequency configuration: c1=6 c2=1, STP = 6hrs, LTP=12hrs.

calls from a SPIT generator efficiently using only this set of configuration parameters. Applying APMG will never block the normal user but will block the calls from a SPIT generator very efficiently.

Figure 3 depicts a scenario where the PMG algorithm blocks calls from a SPIT generator very efficiently but at the risk of blocking a legitimate normal user at some point of time. However, if Adaptive PMG is applied, you can see from the above graph that the normal user would never be blocked but the calls from SPIT generator would be blocked very efficiently.

8.3 Comparing PMG and Adaptive PMG

The results published in this section, verifies that the Adaptive PMG approach works more efficiently than the original PMG algorithm itself. The Adaptive PMG also simplifies the job of an administrator without having to set different policies for different users or user groups and eliminates the need for complex policy management. It dynamically learns the instantaneous behavior of the current caller and applies the corresponding configuration parameters automatically to the PMG.

9 Conclusions

In this paper, we have proposed an approach to fine tune the performance of the PMG algorithm using the concept of frequency detection by introducing two new parameters: short and long term call density thresholds. If the incoming caller exceeds the short or long term call density threshold, he will be treated as a high frequency user and a set of configuration parameters that best suit to block the spammer will be applied at that particular instant. Otherwise the user will be treated as a low frequency user and a set of configuration parameters that best suit the normal users will be applied. Thus Adaptive PMG provides an automated way to apply multiple configuration parameters to the PMG based on the user's instantaneous behavior to combat SPIT more efficiently. We have developed a plug-in that works on top of the PMG algorithm to fine tune its performance. Our experiment results demonstrate the values of configuration parameters that best suit the normal users and spammers.

Our experiment results in section 8 confirm that using our fine tuning approach, the Adaptive PMG works more efficiently than the original PMG algorithm. We believe that the Adaptive PMG could prove itself to be a powerful weapon to combat SPIT when used with strong user authentication.

References

1. Shin, D., Ahn, J., Shim, C.: Progressive Multi Gray-Leveling: A Voice Spam Protection Algorithm. IEEE Network 20(5), 18–24 (2006)
2. Dantu, R., Kolan, P.: Detecting Spam in VoIP Networks. In: USENIX SRUTI 2005 Wksp, Cambridge, MA, pp. 31–37 (2005)
3. Materna, B.: SPIT: Bringing Spam to Your Voicemail Box (2006) Accessed March 8, 2007, http://voipforenterprise.tmcnet.com/feature/service-solutions/articles/4009-spit-bringing-spam-your-voicemail-box.htm
4. Graham-Rowe, D.: A Sentinel to Screen Phone Calls Technology. MIT Review (2006) Accessed March 8, 2007, http://www.technologyreview.com/Infotech/17300/page2/
5. AntiSPIT, Eyeball Accessed March 8, 2007, http://www.eyeball.com/technology/anti_spit.html
6. Turing Test to combat phone spam Virus BULLETIN (2007) Accessed March 8, 2007, http://www.virusbtn.com/news/spam_news/2007/01_30.xml
7. SPIT to make up 70% of calls. Accessed March 8, 2007, http://saunderslog.com/2007/01/30/spit-to-make-up-70-of-calls/
8. NEC Develops World-Leading Technology to Prevent IP Phone SPAM NEC (2007) Accessed January 26, 2007, http://www.nec.co.jp/press/en/0701/2602.html

Spatio-temporal Facial Expression Recognition Using Optical Flow and HMM

Gihan Shin and Junchul Chun

Department of Computer Science, Kyonggi University
San 94-6 Yiui-dong, Yongtong-gu, Suwon, Korea
jcchun@kgu.ac.kr

Summary. Vision-based human computer interaction is an emerging field of science and industry to provide natural way to communicate with computer. In that sense, one of the skills is to infer the emotional state of the person based on the facial expression recognition. In this paper, we present a novel approach to recognize facial expression from a sequence of input images using HMM (Hidden Markov Model) and facial motion tracking based on optical flow. Conventionally, in the HMM which consists of seven basic emotional states, it is considered natural that transitions between emotions are imposed to pass through neutral state. However, in this work we propose an enhanced transition framework model which consists of transitions between each emotional state without passing through neutral state in addition to a traditional transition model. For the localization of facial features from video sequence we exploit template matching and optical flow. The facial feature displacements traced by the optical flow are used for input parameters to HMM for facial expression recognition. From the experiment, we can prove that the proposed framework can effectively recognize the facial expression in real time.

1 Introduction

Facial expression analysis in real time is a challenging issue in human computer interaction and affect-sensitive interface design because face and facial expressions can play essential roles in interpersonal communication. Especially, vision-based face motion tracking and facial expression recognition is an attractive input mode for better human computer interaction. However, facial motion tracking is a tough challenge particularly in varying lighting conditions and a moving, clustered background image. Meanwhile, the facial expression recognition is considered a critical work for human-centered interface design and facial expression control.

Since Ekman and Frisen propose six primary emotion [11], a great amount of research on the face and facial expression recognition have been done [1, 2, 3, 5, 6, 12, 15]. The facial expression recognition can be performed from two different types of input images such as still image and video image. When a still image is used, the cost for processing is relatively low rather than using sequential video image. However, it is difficult to recognize the facial expression based on a conventional pattern matching method when the huge facial variation is involved. For the facial expression recognition from video image neural network, optical flow and Hidden Markov Model etc. are exploited [3, 15] since these approach can use the sequential variation of facial expression. One problem to be resolved for effective facial expression recognition in both

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 27–38, 2008.
springerlink.com

cases is to apply a robust head pose estimation algorithm to the process of facial expression recognition since the head orientation is critical for face detection and recognition. We have been working for developing an automated vision-based facial expression control and animation system [4, 7, 10]. For the part of the work, in this paper we present a real-time approach for inferring emotion from facial expression recognition from video images based on optical flow and HMM. Our work focuses on the design of the classifier used for performing the recognition following extraction of facial features using the real time facial motion tracking system. The proposed method consists of three major phases. In the first phase the work is to detect facial region and localize major facial feature points from the input image. In the second phase the variation of the detected facial feature points from the sequence of the images is traced by utilizing optical flow method. Finally, in the third phase is the facial expression is classified to infer the basic emotion from a person using HMM.

The rest of the paper is organized as follows. Section 2 introduces face and facial feature detection methods. In section 3, the facial motion tracking method based on optical flow is explained. In section 4, HMM-based facial expression recognition is described. In section 5, the experimental results of facial feature tracking under various condition and facial expression recognition are provided. In section 6 conclusion and future works are discussed.

2 Facial Features Localization

The proposed real time facial expression recognition system consists of four major steps: preprocessing, face detection, facial feature extraction and tracking, and facial expression analysis as depicted in Fig 1. From the input video image, enhancement of each frame is performed by histogram equalization and light compensation to improve the quality of the image in the preprocessing step. The appearance of the skin color can change due to varying light conditions. This means the segmentation based on color information can cause misclassification in some cases. In order to minimize the effect of lighting variations we utilize the reference white-level which is the level corresponding

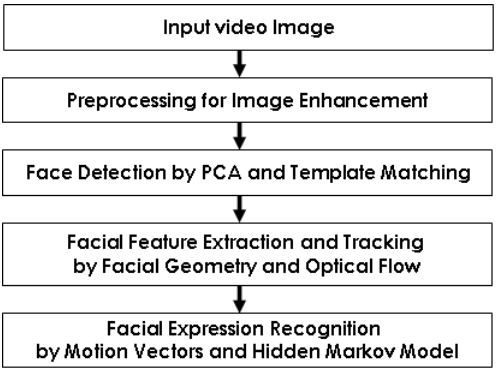


Fig. 1. The steps for proposed spatio-temporal facial expression recognition system

to the specified maximum excursion of the luminance signal in the white direction to normalize the color appearance.

Once a candidate facial region is detected by nonparametric HT(Hue-Tint) skin color model from input video image, the exact face is determined by using PCA(principal component analysis) and template matching technique. From the detected face, the major facial feature points are extracted based on the information of facial geometry and template matching for each subregion of the face. Those facial feature points are traced by optical flow from the sequence of input images. Finally the motion vectors of each facial feature point are used for facial expression recognition by exploiting HMM.

2.1 Face Detection by Skin Color Model

The face detection is first step before tracking the varying facial motion from the sequential input images. In general, color information is known to be efficient for identifying skin region. However, in computer vision, since every color spaces have different properties, color space selection is a very important job for face detection. Therefore, we need to consider some criteria for selecting an efficient skin color models: how to separate color information with chrominance and luminance data, how to describe a chrominance data as complex shaped distributions in a given space, and how to evaluate the amount of overlap between the skin distributions and non-skin distributions in the color space. We proposed nonparametric HT skin color model to detect facial area efficiently rather than using conventional parametric skin color models [7]. With the HT skin color model we can extract the candidate facial region with a minimal overlap between the skin distributions and non-skin distributions. The Fig 2 illustrates the overall steps for face detection.

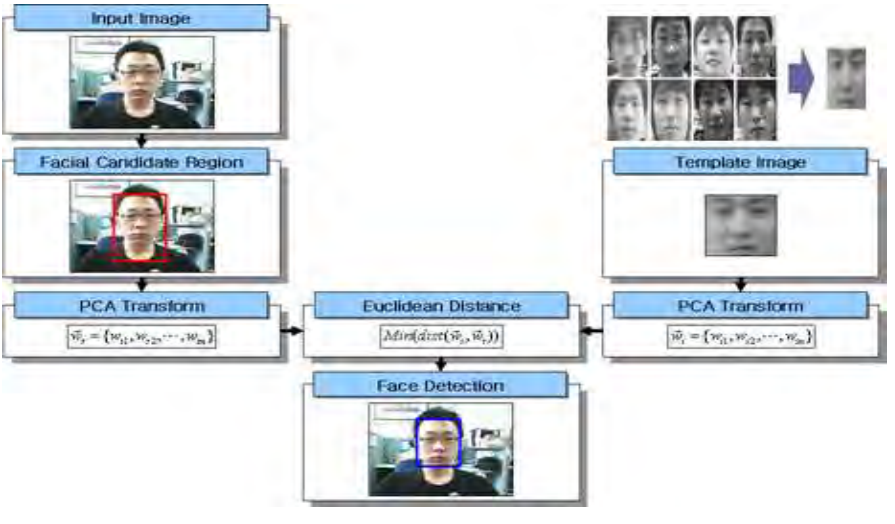


Fig. 2. Major steps for face detection from input image

The exact face is determined by template matching between a given template face image(T) and the candidate facial region(I). A template face image, which is the size of 80×80 is an average face image from training images. Subsequently principal component analysis (PCA) [14] is applied to the average face image to obtain the facial feature vectors. In template matching the feature vectors of template image and those of candidate facial regions are compared to detect the exact face from input image based on L_2 norm defined by

$$Distance(\omega_{\mathbf{I}}, \omega_{\mathbf{T}}) = \sqrt{\sum_{n=0}^{79} (\omega_{\mathbf{I_n}} - \omega_{\mathbf{T_n}})}, \tag{1}$$

The proposed face detection approach can detect facial region efficiently, even though various conditions such as light variation, complex back ground image and face image with glasses are involved.

2.2 Facial Feature Extraction Method

The major facial feature vectors to be used for facial expression analysis must be extracted from the detected face. In this work, we detect only 18 major feature points, which are defined in MPEG 4, for facial expression recognition. The standard proportion for the human face can be used to determine its proper feature position and find their orientation. In this work, the detected face image is separated into two local regions which include eye or lip area by using the information of facial geometry. The eyes and eyebrow are located in 2/3 of the facial geometry and lip is in 1/3 of the face as shown in Fig 3. This process minimizes the search area of major facial feature points. Once each candidate local regions which include the described facial feature points are selected, then the canny edge detection method is applied to each region to determine the edges of eyes, lip and eyebrow. Subsequently, template matching is used to detect facial features. When the proposed method is used, the exact positions of eyes can be extracted even from a person who wears glasses.

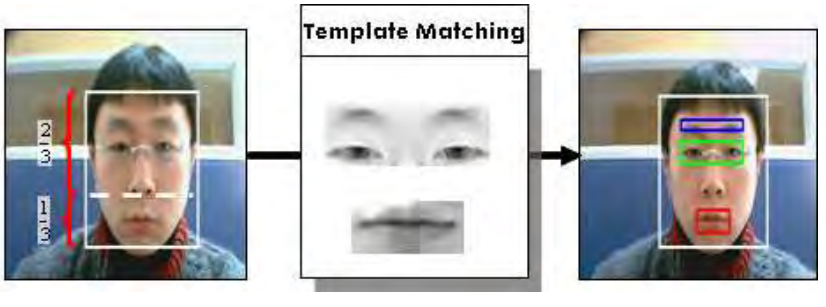


Fig. 3. Facial geometry and templates image for eye, eyebrow, mouth region for facial feature extraction

3 Facial Motion Tracking

The detected facial feature vectors, which will be used for facial expression recognition, are tracked by optical flow method [9]. The optical flow is a method to estimate the motion of brightness pattern between the two sequential frames. In this paper, we adopt Lucas-Kanade optical flow which is robust to noise and highly accurate for motion tracking. In general optical flow is expressed by

$$I(x,t) = I(x+\mu, t+dt) \tag{2}$$

In formula (2) **I** is brightness, **x** is the location of pixel, μ is the change ratio of pixel, **t** is time. Based on Taylor series, the equation (2) can be redefined by

$$I(x+\mu, t+dt) = I(x,t) + \bigtriangledown I \cdot \mu + \frac{\partial I}{\partial t} \cdot dt \tag{3}$$

In formula (3) $\bigtriangledown$**I** means the ratio of the spatial image change. From the formula (3), constraints formula of the optical flow can be reduced and expressed as follow.

$$\bigtriangledown I \cdot \mu + \frac{\partial I}{\partial t} \cdot dt = 0 \tag{4}$$

Lucas-Kanade optical flow algorithm can be expressed as follow for the window ω by minimizing the value of the left-hand term of (4)

$$\sum_{x \in \omega} \omega^2(x)(\bigtriangledown I \cdot \mu + \frac{\partial I}{\partial t} \cdot dt) \tag{5}$$

Fig 4 illustrates the results of the facial features tracking by the optical flow method.

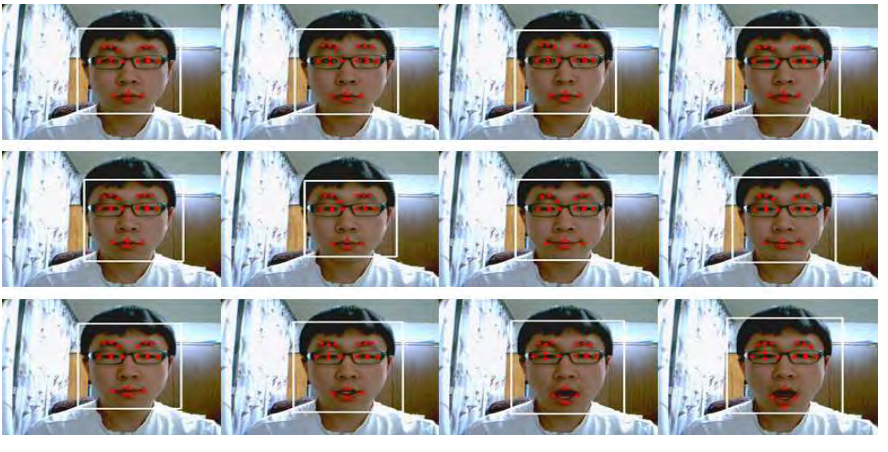


Fig. 4. Examples of facial feature variation traced by using optical flow

4 HMM-Based Facial Expression Analysis

The next step is to recognize the facial expression based on the facial feature points traced from video images. For recognizing the facial expression, we use Hidden Markov model, which have been widely used for many classification and modeling problem. The strong point of HMM is its ability to model non-stationary signals or events even though its time-consuming approach to find the best match for the classification. The HMM uses the transition probabilities between hidden sates and learns the conditional probabilities of the observations if the state of the model is given. In inferring emotion from the facial expression recognition, the signal is the measurements of motion vectors between the emotional states. The vectors are non-stationary since an facial expression can be displayed at varying rates with varying intensities [3].

In this paper the HMM consists of five basic facial emotional states: happy, angry, surprise, sad, and neutral state. Since a certain facial expression from the sequence of image is represented by a temporal sequence of facial motion, each expression can be modeled by an HMM trained for the particular type of each expression. Such an HMM can be defined as follow:

$$\lambda = (A, B, \pi) \tag{6}$$

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), i \leq i, j \leq N \tag{7}$$

$$B = \{b_j(O_t)\} = P(O_t | q_t = S_j), i \leq j \leq N \tag{8}$$

$$\pi_j = P(q_1 = S_j) \tag{9}$$

Where **A** is the state transition probability matrix, **B** is the observation probability distribution, π is the initial state distribution, and **N** is the number of the states of the HMM. The observation $\mathbf{O_t}$ can be either discrete or continuous, and can be vectors, which represents continuous motion of the facial feature points. Thus, **B** is represented by the probability density function of the observation vector at time **t** given the state of the model. The Gaussian distribution is chosen to represent the probability density function and is defined by

$$B = b_j(O_t) \sim (\mu_j, \Sigma_j), \;\; i \leq j \leq N \tag{10}$$

where μ_j and Σ_j are the mean vector and full covariance matrix, respectively. The parameters(λ) of the model of emotion-specific HMM are learned by using Baum-Welch formulas [13], which drives the maximum likelihood estimation of the model parameters. Given an observation sequence $\mathbf{O_t}$, where $t \in (1, T)$, the probability of the observation given each seven expression models $P(O_t|\lambda_j)$ can be computed by the forward-backward procedure [8]. Finally, the sequence is classified as the emotion corresponding to the model that yields the maximum probability as follow

$$c^* = argmax[P(O|\lambda_c)], \;\; 1 \leq c \leq 5 \tag{11}$$

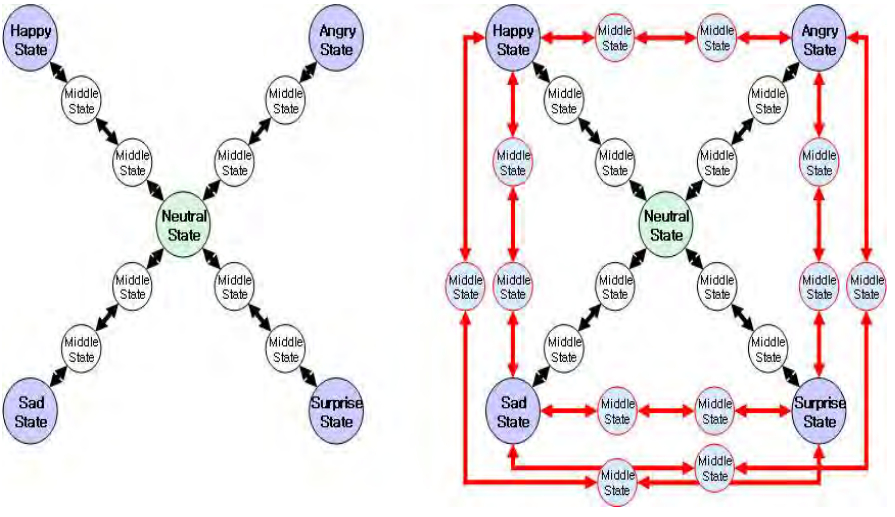


Fig. 5. Comparison between the conventional state transition model(left) and enhanced state transition model(right)

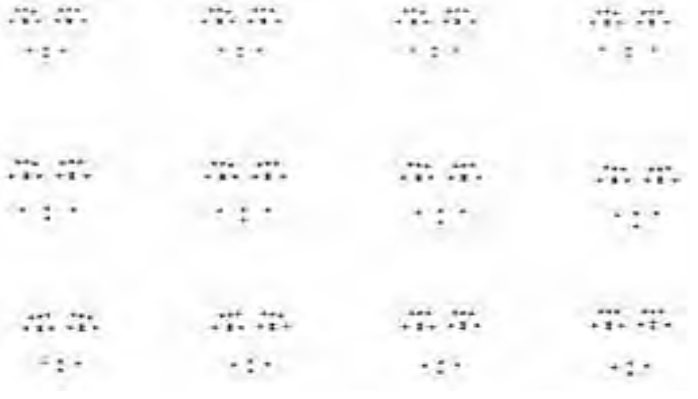


Fig. 6. Examples for intermediate data between specific emotional states which is generated by interpolation(from top to bottom (a) Interpolated data between neutral and happy states (b) Interpolated data between happy and angry states (c) Interpolated data between sad and surprise states

Conventionally, in HMM which consist of seven emotional states, it is considered natural that transitions between emotions are imposed to pass through neutral state. However, in this work we propose an enhanced transition framework model which consists of transitions between each emotional state without passing through neutral state in addition to a traditional transition model. The transition model is shown in Fig 5.

Most computational overhead resides in the training phase. In the existing HMM based facial expression recognition, the training data are obtained from the selected frames between the emotional states. However, in this work, the training data of intermediate state can be created by using the interpolation of the facial feature points of both basic emotional states. Therefore, we can reduce the computational cost in training process. Fig 6 illustrates some examples of the intermediate state between both emotional states by the interpolation of facial feature points. Based on the proposed model, we can effectively infer the emotion of the in-between frames between two extreme expressional states for instance a transition between happy state and angry state when neutral state is not involved.

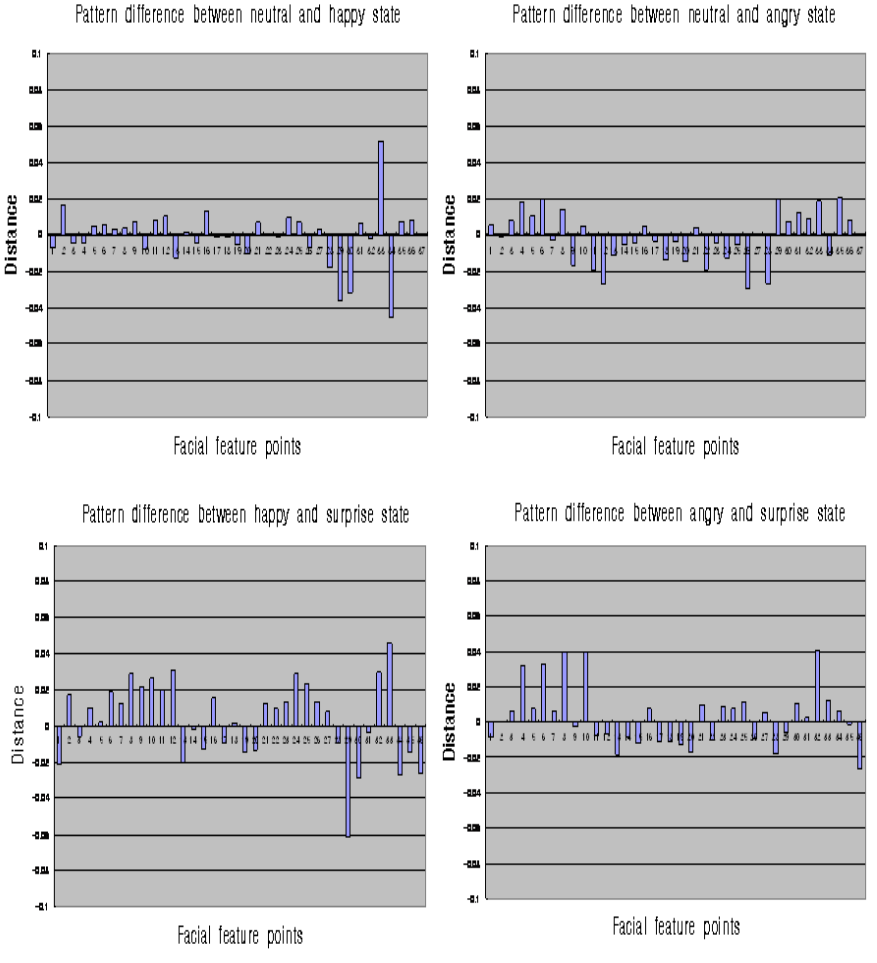


Fig. 7. Motion pattern of feature points between two different expression states (from top to bottom clockwise, neutral-happy, neutral-angry, happy-surprise, angry-surprise)

The feature motion pattern between two different expressional states can be evaluated by obtaining the distance between the corresponding feature points of each peak state representative of the expression. For every 18 facial feature points of two expressional states, the distance vector **d** is considered as a motion pattern of a specific feature point. When coordinates of corresponding feature point of two different expression state are $\mathbf{e_i}(\mathbf{x_k}, \mathbf{y_k})$ and $\mathbf{e_j}(\mathbf{x_k}, \mathbf{y_k})$, the distance between two states is defined by

$$d_i = (e_i(x_k) - e_j(x_k),\ e_i(y_k) - e_j(y_k)),\ k = 1 \cdots 18 \tag{12}$$

Some examples of the motion patterns between two peak frames of representative expressional states are illustrated in Fig 7. These motion patterns are used for the classification of expressional state

5 Experimental Results

From the experiment we can prove that the proposed facial feature localization method can correctly extract major facial features as illustrated in Fig 8.

Since the tracking of facial feature variation is very important in facial expression recognition, the results of facial feature extraction is critical in overall process. As shown in Fig 9, the 24 sequential frames which involve the state transitions from neutral, happy state, surprise state, angry state to sad state are used for the facial expression recognition. The first two frames of the sequential images are apparently neutral states and the other frames are one of other expressional states.

The recognition ratio between two different emotional states is describes in table 1. The graph in Fig 10 depicts the results of facial expression recognition based on both the conventional state transition HMM model and the proposed transition HMM model. When the conventional HMM model is used for expression recognition, the neutral

Fig. 8. Results of facial feature detection from persons with glasses (top: previous method, bottom: proposed method)

Fig. 9. A sequence of face images: from top to bottom and left to right facial expression variation with neutral, happy, surprise, angry and sad

Table 1. The recognition ratio between two different emotional states

Emotional States	Neutral	Happy	Anger	Sad	Surprise
Neutral	none	74.94	80.46	71.13	79.89
Happy	77.09	none	76.69	80.54	80.72
Anger	79.59	73.90	none	79.29	77.38
Sad	78.22	74.45	81.84	none	76.87
Surprise	78.70	73.93	83.00	78.06	none

states should be always detected in-between the two different expressional sates. The states marked by a circle are the neutral states. The frames weakly coupled with a specific facial expression are classified into the neutral state even though those frames are

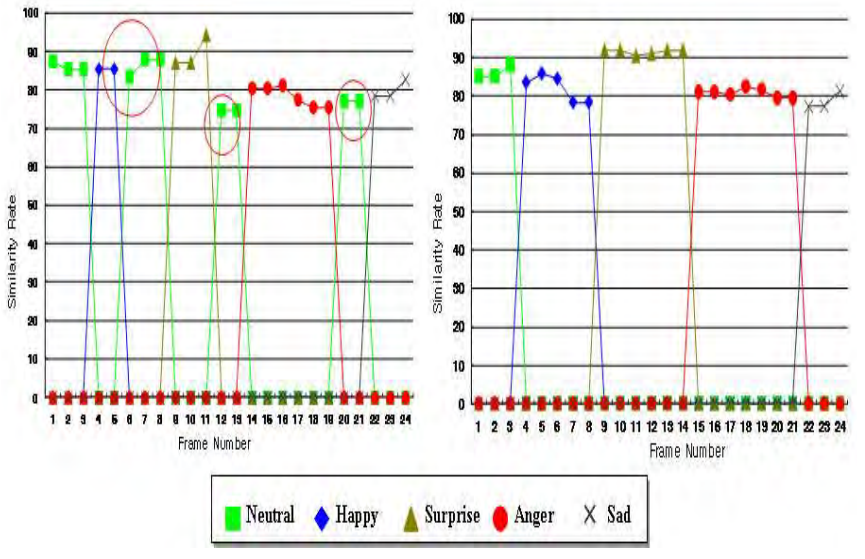


Fig. 10. Results of facial expression recognition using the sequence of facial images illustrated in Fig 9. A result by conventional HMM (left), a result by the proposed model (right).

involved in the transitional state between two extreme emotional states. However, the proposed approach can apparently classify the frames which impose a certain expressional state, which is not a neutral state but weakly coupled with a certain basic state, into a specific emotional sate rather than classifying them into a neutral state.

6 Conclusion and Future Works

In this paper, we propose a robust approach to detect face, localize facial feature points, track facial motion and recognize the facial expression from a sequence of input face images. For this purpose, we utilize optical flow method for tracking facial features and enhanced state transition HMM for analyzing facial expression. From the experiments, we can show the proposed method can effectively detect facial feature points and recognize the facial expression from sequences of input images in real time basis. As for the future work, we will combine the head pose estimation to recognize the facial expression. In addition, the recognized facial expression will be applied to control the facial expression of a 3D avatar in producing vision-based computer animation.

Acknowledgement. This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy(MKE) 21st Century Frontier R & D Program in Korea and a result of subproject UCN 08B3-O3-40S.

References

1. Fasel, B., Luettin, J.: Automatic Facial Expression Analysis: A Survey. Pattern Recognition 36(1), 259–275 (2003)
2. Chien, C.C., Chang, Y.J., Chen, Y.C.: Facial Expression Analysis Under Various Head Poses. In: Proceedings of 3rd IEEE Pacific-Rim Conf. on Multimedia, pp. 16–18 (2002)
3. Cohen, I., Sebe, N., Garg, A., Chen, L.S., Huang, T.S.: Facial expression recognition from video sequences: temporal and static modeling. Computer Vision and Image Understanding 91, 160–187 (2003)
4. Chun, J.C., Kwon, O.R., Park, P.: A Robust 3D Face Pose Estimation and Facial Expression Control for Vision-Based Animation. In: Cham, T.-J., Cai, J., Dorai, C., Rajan, D., Chua, T.-S., Chia, L.-T. (eds.) MMM 2007. LNCS, vol. 4351, pp. 700–708. Springer, Heidelberg (2007)
5. Huang, J., Blanz, V., Heisele, B.: Face Recognition with Support Vector Machines and 3D Head Models. In: International Workshop on Pattern Recognition with Support Vector Machines, Niagara Falls, Canada, pp. 334–341 (2002)
6. Jonsson, K., Matas, J., Kittler, J., Li, Y.P.: Learning Support Vectors for Face Verification and Recognition. In: Procedings of 4th IEEE International Conference on AFGAGF 2000, pp. 208–213 (2000)
7. Min, K.P., Chun, J.C., Park, G.R.: A Nonparametric Skin Color Model for Face Detection from Color Images. In: Liew, K.-M., Shen, H., See, S., Cai, W. (eds.) PDCAT 2004. LNCS, vol. 3320, pp. 115–119. Springer, Heidelberg (2004)
8. Rabiner, L.R.: A Tutorial on Hidden Markov Models and selected Applications in Speech Processing. Proc. IEEE 77(2), 257–286 (1989)
9. Black, M.: Robust incremental optical flow, PhD thesis. Yale University (1992)
10. Kwon, O.R., Chun, J.C., Park, P.: Cylindrical Model-Based Head Tracking and 3D Pose Recovery from Sequential Face Images. In: IEEE International Conference on Hybrid Information Technology, pp. 135–139 (2006)
11. Ekman, P., Friesen, W.V.: Facial Action Coding System (FACS). Consulting Psychologist Press, Inc. (1978)
12. Michel, P., El Kaliouby, R.: Real Time Facial Expression Recognition in Video using Support Vector Machines. In: ICMI 2003, pp. 258–264 (2003)
13. Levinson, S.E., Rabiner, L.R., Sondhi, M.M.: An Introduction to the Application of the Theory of Probilitic functions of a Markov Process to Automatic Speech Recognition. Bell Lab System Technical J. 62(4), 1035–1072 (1983)
14. Zhu, Y., De Silva, L.C., Ko, C.C.: A Solution for Facial Expression Representation, and Recognition. Signal Processing: Image Communication 17, 657–673 (2002)
15. Zhu, Y., De Silva, L.C., Ko, C.C.: Using Moment Invariants and HMM in Facial Expression Recognition. Pattern Recognition Letters Archive 23, 83–91 (2002)

An Approach for Anomaly Intrusion Detection Based on Causal Knowledge-Driven Diagnosis and Direction

Mahmoud Jazzar[1] and Aman Jantan[2]

[1] School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia
mahmoudj@cs.usm.my
[2] School of Computer Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang, Malaysia
aman@cs.usm.my

Summary. Conventional knowledge acquisition methods such as semantic knowledge, production rules and question answering systems have been addressed to a variety of typical knowledge based systems. However, very limited causal knowledge based methods have been addressed to the problem of intrusion detection. In this paper, we propose an approach based on causal knowledge reasoning for anomaly intrusion detection. Fuzzy cognitive maps (FCM) are ideal causal knowledge acquiring tool with fuzzy signed graphs which can be presented as an associative single layer neural network. Using FCM, our methodology attempt to diagnose and direct network traffic data based on its relevance to attack or normal connections. By quantifying the causal inference process we can determine the attack detection and the severity of odd packets. As such packets with low causal relations to attacks can be dropped or ignored and/or packets with high causal relations to attacks are to be highlighted.

Keywords: Intrusion detection, False alerts, Fuzzy cognitive maps, Security.

1 Introduction

The rapid advances and spreads of network-based technologies in both private and government sectors have managed in increasing network attacks which results in less secure systems. On the other hand, the lack of authentication means, unfortunately, results in limitless possibilities of malicious intrusions, therefore secure computer networks remain an issue for local/private and national/global security.

Intrusion Detection Systems (IDS) main concern is to function as a monitor to the network state by looking for unauthorized usage, denial of service and anomalous behavior [4]. Such a system needs to be efficient, configurable and is able to deal with events as they occur. Moreover, IDS main goal is to classify system activities into two major categories: normal and suspicious (intrusion) activities [19]. Typically, network based IDS process system activities based on network data and make a decision to evaluate the probability of action of these data to decide whether these activities are normal or intrusion.

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 39–48, 2008.
springerlink.com

In order to evaluate the system activity and trace the probability of normal vs. intrusive data, the basic knowledge of network attacks is necessary. The problem is that network attacks may not happen at one single action such that one massive attack may start by seemingly innocuous or by small probe actions to take place [16]. Here, we suggest considering the domain knowledge of network data (packets). Thus, we need to extract causal knowledge from network packets and make inference (diagnose) with it and direct normal and abnormal related patterns for further actions.

In this paper, we use the FCM to express causal knowledge of data and calculate the severity/relevance of network data to attacks. Therefore, benign concepts can be dropped or ignored and/or can be addressed as a potential risk of error/attack caused. The rest of the paper is arranged as follows: related works are discussed in section 2, the causal knowledge-driven intrusion detection approach in section 3 and the experimental results and discussions are available in section 4. Finally in section 5, we provide conclusions and recommendations for future research.

2 Related Works

To our best knowledge, existing studies on causal knowledge acquisition for intrusion detection are very limited. However, our study was motivated by a work done recently on an intelligent IDS prototype [17] and the probe detection system (PDSuF) prototype [11]. The proposed intelligent IDS system [17] use fuzzy rule based and FCM as decision support tools and inference techniques. The proposed decision engine analyzes both misuse and anomaly modules information and combine both results for generating the final reports. For misuse information, the decision engine assesses the results from different misuse modules in order to capture misuse scenario. The anomaly detection module information is represented by neural networks such as neurons, weights and relationship between the nodes. In order to generate alerts, suspicious events are treated as causal relations which are generated based on a scenario of combining evidence of multiple suspicious events via FCM.

The probe detection system (PDSuF) prototype [11] uses FCM for intrusion detection. In the proposed system, the decision module uses FCM to capture and analyze packet information to detect SYN flooding attacks using a conventional FCM to measure the effect value based on the weight value between different variable events. Later, the decision module measures the degree of risk of DoS and trains the response module to deal with attacks. However, our approach is different from these approaches in such a manner that the suspicious events are generated from the flow of network packets depending on relevancy factors and causal relations among these factors of network data flow using the FCM framework. Based on the domain knowledge of network data, our FCM framework uses a causal reason to measure the severity on network data.

The various techniques used include data mining [12], AI techniques [5], fuzzy logic [6], neural networks [13], neuro-fuzzy approach [2] and many other machine

learning techniques. These techniques and approaches work on logs/alerts directly and indirectly by building new strategies to tackle intrusions of various types to improve the detection process. The biggest challenge here is to develop an intelligent inference engine model to defense-in depth i.e. able to deal with uncertainty and detect novel attacks with low rate of false alerts. Moreover, any optimal solutions of an adaptive IDS system should provide the means of real-time detection and response as well as high level trust among the IDS components.

3 Our Causal Knowledge-Driven Intrusion Detection Approach

Each network data is a network packet or connection which has certain identities either qualitative or quantitative such that (IP, Port, Flag, time, data, etc.). These identities could either be related or not related to certain attack connection. We consider these identities as concepts to the FCM framework. The FCM module calculates the errors caused out of these concepts and the degree of relevance to certain or measure error (attack).

Thus, later we can estimate how much these concepts are related to attacks. The main advantage here is to call attention to how domain knowledge of neurons (network packets) can contribute on tracing new attacks or find the path of on-going or existing attacks. Each feature parameter of network data is measured based on a comparison criteria to detect interrelation between neurons i.e. determine the attack detection. To calculate the abnormality factor per packet we need to estimate the effect value of each feature parameter. The total degree of abnormality of odd neurons is calculated according to the following factors, as summarized in Table 1.

Using factors, rules and effect values in Table 1, we can estimate the total degree of abnormality per packet according to the following rule:

$$Un(x) = \sum_{i=1}^{n} E_i \tag{1}$$

where $Un(x)$: Anbormality per packet
E_i: Effect value of packet
n: Total feature number of abnormality

Once the abnormalities per un-clustered packets are calculated, the low malicious packets are dropped or ignored and the rest are considered as concepts in the FCM. It is now important to measure the effect/inference value among the suspicious concepts to determine the path of the existing or ongoing attack. If the effect value is zero then there is no relationship among these concepts. Table 2 shows the total degree of effect value and relations between neurons.

Table 1. Factors, rules and effect value

Factor	Factor Rule	Effect (E)
Availability	$\begin{cases} 1 & X \in S \\ 0 & X \notin S \end{cases}$ Where X: Comparison S: set of features	0.1
Similarity	$\begin{cases} 1 & X = S_i \\ 0 & X \neq S_i \end{cases}$ Where S_i is a feature of set S and X is comparison	0.1
Occurrences	$\log_2 \frac{1}{p(x)}$ Where $p(x)$ is x's probability	0.2
Relevancy	$\frac{MaxF_i(x)}{\sum_{x \in s} F_i(x)}$ Where x : comparison, $MaxF_i(x)$ is the maximum frequency of occurrences, and $\sum_{x \in s} F_i(x)$ is the total sample size (number of trials)	0.2
Independency	$P(x)P(y)$ Where $P(x)$ is the x's probability and $P(y)$ is the y's probability	0.2
Correlation	$Cov(X_t Y_t)/S_{X_t} S_{Y_t}$ is the covariance of X and Y comparisons at time t and the standard deviation	0.2

Table 2. Effect and relation value trace

Normal	0
Slight	0.2
Low	0.4
Somehow	0.6
Much	0.8
High	1

3.1 FCM Procedure

FCM are a soft computing modeling techniques generated from the combination of fuzzy logic and neutral networks [3, 9, 10, 18]. FCM consist of nodes (concepts) and causal relations between the nodes formed in a structured collection (graph). The structure can be presented as an associative single layer neural networks which work on unsupervised mode whose neurons are assigned to concepts meanings and the interconnection weights represent the relationship among these concepts.

According to [1] in the FCM model, the directional influences are presented as all-or-none relationships i.e. FCM provide qualitative as oppose to quantitative information about relationships. In this work, the task of FCM is to determine the casual relation between the suspicious or odd neurons noted or identified to quantify the causal inference process. By quantifying the causal inference process we can determine the attack detection and the severity of odd neurons. As such, neurons with low casual relations can be dropped i.e. reduce the false alerts. The following steps are the general FCM procedure.

1. Define the number of odd neurons (concepts)
2. Calculate the abnormality per neuron
   a) Drop neuron if the abnormality is low
3. Call FCM initialization
4. Call FCM Simulation

The number of neurons includes all those unidentified and attack neurons. At every epoch we process 100 neurons from the data set, which later we pick the alert and non-alert neurons and calculates the abnormality per each and every neuron to drop the low attack related and consider the rest as concepts for the FCM framework.

FCM Initialization

Initializing the FCM includes the definition of the FCM concepts and building the relations among these concepts by building a global matrix which can be calculated according to [9, 10]. However, in order to build that matrix we define the weight of odd neurons according to the total effect factor $Un(x)$ and the grade of causality W_{ij} between the nodes C_i and C_j according to the following assumptions:

1. If $C_i \neq C_j$ and $E_{ij} > 0$ then ${W_{ij}}^{+} = \max\{E_{ij}^t\}$
2. If $C_i \neq C_j$ and $E_{ij} < 0$ then ${W_{ij}}^{-} = \max\{E_{ij}^t\}$
3. If $C_i \neq C_j$ then $E_{ij} = 0$ and W_{ij} is zero

FCM Simulation

Once the FCM is constructed, it is important now to measure the overall simulation of the system which consists of s input states such that $M = \{s_1, s_2, \ldots s\}$

where $s_i \in [0, 1]$. After n number of iterations the output is $\overline{M}$ i.e. the predictions of the FCM model. The simulation of FCM follows the following steps:

1. Read from input state
2. Calculate the Effect factors
   a) Drop low effect factors
3. Until the system convergence
   a) Show the link of related factors

4 Experimental Results and Discussion

A causal knowledge-driven intrusion detection model is a defense-in-depth network based intrusion detection scheme. The model utilizes the domain knowledge of network data to analyze the packet information. Based on the analysis given, benign packets are dropped and high risk packets can be highlighted or blocked using a causal knowledge reason in FCM. The flowchart of the detection module is illustrated in Figure 1.

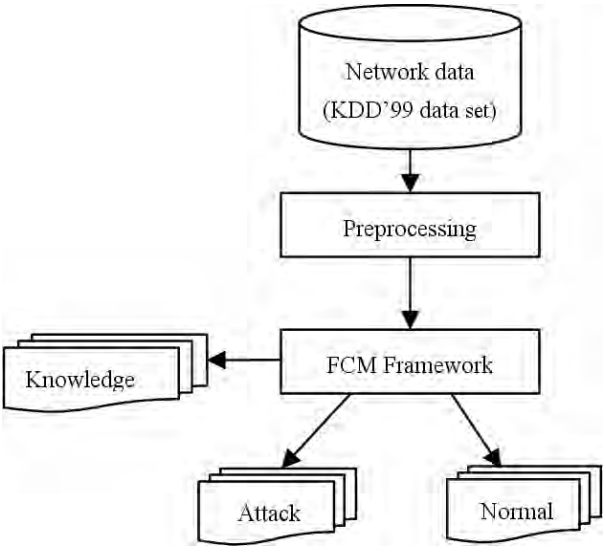


Fig. 1. Flowchart of the detection module

In this experiment, we use the most popular IDS evaluation data in which most of the researchers were aware of and used for evaluating their research, the KDD Cup 1999 intrusion detection contest data [7] followed by the success of the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs (MIT Lincoln Laboratory) [14]. The aim of DARPA evaluation was to assess the current state of Intrusion Detection Systems at the Department of Defense at the

Table 3. Selected data set records

Attack Category	Attack Name	# Records	Total
Normal	–	1020	1020
DoS	Neptune	105	367
DoS	Smurf	124	
DoS	Back	42	
DoS	Land	40	
DoS	Pod	33	
DoS	Teardrop	23	
Probe	Ipsweep	79	319
Probe	Nmap	59	
Probe	Portsweet	77	
Probe	Satan	44	
Probe	Mscan	36	
Probe	Saint	24	
U2R	buffer_overflow	82	217
U2R	Sqlattack	79	
U2R	Perl	8	
U2R	Xterm	22	
U2R	Rootkit	26	
R2L	guess_password	41	97
R2L	Imap	2	
R2L	ftp_write	22	
R2L	Phf	20	
R2L	Sendmail	12	

U.S. by simulating a typical U.S. Force LAN. However, Lincoln Labs acquired 9 weeks of raw data collection for the evaluation.

The collected raw data processed into connection records, about 5 million of record connection. The data set contain 41 attributes for each connection record plus one class label and 24 attack types which fall into four main attack categories [15] as follows:

1. Probing: surveillance attack categories
2. DoS: denial of service
3. R2L: unauthorized access from a remote machine
4. U2R: unauthorized access to local super user (root) privileges

The data set was established to evaluate the false alarm rate and the detection rate using the available set of known and unknown attacks embedded in the data set [8]. We select a subset for testing our experiment. The selected subset contains 2020 records with non zero values (as shown in Table 3) because some attacks are represented with few examples and the attack distribution in the large data set is unbalanced. However, collection, preprocessing and calculation of false and true alert of test data are followed as in [16]. We implement and run

our experiment on a system with 2.667GHz Pentium4 processor 506 and 256MB PC3200 DDR400 RAM running windows XP.

FCM Framework

In this module, the received data attributes will be carried for fine-tuning in the FCM framework as discussed in section 3 and 3.1. In this module, neurons which represent low effect or less correlated to other attack like neurons are dropped or ignored and the high suspicious nodes are highlighted.

In our experiment, the performance measure of our causal knowledge-driven intrusion detection approach are carried out solely on the selected data subset from the corrected .gz file of the KDD'99 data set [7] which contains test data with corrected labels. For instance, we calculate the detection rate and the false alarm rate according to [16] the following assumptions:

FP: the total number of normal records that are classified as anomalous
FN: the total number of anomalous records that are classified as normal
TN: the total number of normal records
TA: the total number of attack records
Detection Rate = $[(\mathrm{TA} - \mathrm{FN})/\mathrm{TA}] \times 100$
False alarm Rate = $[\mathrm{FP}/\mathrm{TN}] \times 100$

The experimental results show that causal seasoning is a vital approach for intrusion detection. We believe that, further improvement on the structure of FCM will improve the detection accuracy and expose more information about the attack details. Table 4 and Table 5 show the experimental results obtained.

Table 4. Experimental results

Attack type	**# Records**	**# Detection Records**
Normal	1020	1016
Probe	319	290
DoS	367	360
U2R	217	183
R2L	97	71
Overall	2020	1920

Table 5. Detection accuracy and false alarm rate

Detection rate	**False alarm rate**
90.5%	9.7%

5 Conclusion

As a conclusion, we have described a causal knowledge-driven intrusion detection approach using the FCM. Initial result revealed that causal knowledge diagnosis and direction on network data flow is a vital approach for anomaly intrusion detection. For future research, experiment should be done on real time traffic data and investigating methods for proper feature selection and presentation.

References

1. Aguilar, J.: A dynamic fuzzy-cognitive-map approach based on random neural networks. International Journal of Computational Cognition 1(4), 91–107 (2003)
2. Alshammari, R., Sonamthiang, S., Teimouri, M., Riordan, D.: Using neurofuzzy approach to reduce false positive alerts. In: Proceedings of Fifth Annual Conference on Communication Networks and Services Research (CNSR 2007), pp. 345–349. IEEE Computer Sociey Press, Los Alamitos (2007)
3. Axelrod, R.: Structure of Decision: The Cognitive Maps of the Political Elites. Princeton University Press, New Jersey (1976)
4. Denning, D.E.: An intrusion model. IEEE Transactions on Software Engineering SE-13(2), 222–232 (1987)
5. Depren, O., Topallar, M., Anarim, E., Kemal, C.M.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications 29, 713–722 (2005)
6. Dickerson, J.E., Juslin, J., Koukousoula, O., Dickerson, J.A.: Fuzzy intrusion detection. In: Proceedings of IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, Vancouver, British Columbia (2001)
7. KDD Cup 1999 Data. Knowledge Discovery in Databases DARPA Archive. Accessed December 2007, `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
8. Kendall, K.: A database of computer attacks for the evaluation of intrusion detection systems. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA (1999)
9. Kosko, B.: Fuzzy cognitive maps. International Journal of Man-Machine Studies 24, 65–75 (1986)
10. Kosko, B.: Fuzzy Engineering. Prentice-Hall, Englewood Cliffs (1997)
11. Lee, S.Y., Kim, Y.S., Lee, B.H., Kang, S., Youn, C.H.: A probe detection model using the analysis of the fuzzy cognitive maps. In: Proceedings of the International Conference on Computational Science and its Applications (ICCSA), vol. 1, pp. 320–328 (2005)
12. Lee, W., Stolfo, S.J., Mok, K.W.: Adaptive intrusion detection: A data mining approach. Artificial Intelligence Review 14(6), 533–567 (2000)
13. Liu, Y., Tian, D., Wang, A.: ANNIDS: Intrusion detection system based on artificial neural network. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an (2003)
14. MIT Lincoln Lab: DARPA Intrusion Detection Evaluation Plan. Accessed December 2007, `http://www.ll.mit.edu/IST/ideval/data/2000/2000_data_index.html`

15. Peddabachigari, S., Abraham, A., Grosan, C., Thomas, J.: Modelling intrusion detection system using hybrid intelligent systems. Journal of Network and Computer Applications (2005), DOI 10.1016/j.jnca.2005.06.003
16. Sarasamma, S.T., Zhu, Q.A., Huff, J.: Hierarchal kohonenen net for anomaly detection in network security. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 35(2), 302–312 (2005)
17. Siraj, A., Vaughn, R.B., Bridges, S.M.: Intrusion sensor data fusion in an intelligent intrusion detection system architecture. In: Proceedings of the 37th Hawaii International Conference on System Sciences (2004)
18. Stylios, C.D., Groumpos, P.P.: Mathematical formulation of fuzzy cognitive maps. In: Proceedings of the 7th Mediterranean Conference on Control and Automation (MED 1999), Haifa, Israel (1999)
19. Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer Communications 30, 2201–2212 (2007)

Challenges Relating to RFID Implementation within the Electronic Supply Chain Management – A Practical Approach

Belal Chowdhury[1], Morshed U. Chowdhury[2], and Clare D'Souza[3]

[1] La Trobe University, Melbourne 3086, Australia
bchowdhury@psolution.com.au
[2] Deakin University, Melbourne, Victoria 3125, Australia
muc@deakin.edu.au
[3] La Trobe University, Melbourne 3086, Australia
C.Dsouza@latrobe.edu.au

Abstract. The use of RFID (Radio Frequency Identification) technology can be employed for not only reducing companies management costs but also to track uniquely each shipping container, pallet, case, and product being manufactured, shipped and sold, to increase visibility and accountability in the supply chain. RFID technology connects the supply chain players (i.e., suppliers, manufacturers, distributors, retailers and customers) and allows them to exchange data and product information. Despite these potential benefits, there are challenges and obstacles with the deployment of a RFID-enabled system in the global supply chain. The paper outlines the major RFID issues faced by supply chain management. In this paper, we also present a case study on pharmaceutical supply chain management (SCM) applications by addressing and examining the issues of RFID implementation in a SCM system.

Keywords: RFID, SCM, and PSCMS.

1 Introduction

A supply chain is a network of facilities that encompasses all of the activities and information flow necessary for the transformation of goods from the raw material to finished products, and the distribution of these finished products to the end user. It is regarded as a support system that is used to connect the members of a value chain in a proficient network of relationships. Supply chains if appropriately managed can decrease costs, develop customer service, improve the enterprise's knowledge base, increase competence within any organization and combat competitors [1].

Supply chain management (SCM) drives and enhances those businesses that rely heavily upon logistics for moving products around [2]. Current supply chain comprise of processes that include how items move between supplier, manufacturer, wholesaler, and retailer and eventually reach the end user. SCM encompasses not only domestic and international industry sectors but also suppliers. Increasing global competition is putting pressure on supply chains to be cost effective, efficient and more proactive to gain competitive advantage in the market. There is a growing trend amongst businesses and governments in the developed as well as emerging economies such as China, and India to co-ordinate and increase collaboration in their governments, business activities

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 49–59, 2008.
springerlink.com

and processes. Effective and well-planned electronic SCM systems within governments and industry provide opportunities to continuously improve operations both within and external to an organization [3]. In recent years, SCM and information technology have gained significant importance aided by anecdotal success stories of industries obtaining competitive advantage using these systems. Information technology is contributing a significant role in bringing opportunities and challenges to SCM and making it grow at an even faster pace.

In the last few years supply chain players (e.g., manufacturers) are persistently struggling to get the right products to the right retailers at the right time [4]. We propose that in order to be functionally proficient, Radio Frequency Identification (RFID), which is the emerging and latest data capture technology of the twenty-first century is likely to solve this problem and to make the largest impact on SCM over the next decades. It enables companies to automatically identify and track items in the supply chain. RFID besides allowing for supply chain automation to stamp out counterfeit products such as drugs, fight terrorism, and at the same time help companies like Wal-Mart keep its shelves stocked [5]. RFID-based SCM systems also uniquely identify every product in real-time across the supply chain to increase efficiency in areas like retail, hospitals, farming, and public transport. They connect suppliers, manufacturers, distributors, retailers and customers and allow them to exchange product and trading partner data. As a result companies can make substantial annual cost savings by reducing inventory levels and lowering distribution and handling costs, increasing security and product integrity, and improving flexibility [4].

RFID-based systems in SCM use tiny chips, called tags (or smart tags) contain and transmit product or item information to a RFID reader, a device that in turn interfaces/integrates with the company's IT systems for processing through wireless communication (i.e., air interface). The product information such as the product ID, manufacture date, price, and its distribution point can be written to the tag to enable greater product accountability and safety. Due to the larger amounts of data storage and capacity for interactive communication RFID technologies are far more powerful than the conventional identification techniques such as barcodes. In addition they provide unique identification for each tagged unit whereas barcodes are identical for every unit of the same product. Unlike barcodes, RFID technologies do not need line of sight and the tag (RFID) can be read without actually seeing it [6]. Also, the RFID tag's read rate is much faster than the barcode system. Some advanced RFID readers that can read up to 60 different RFID tags at approximately the same time, while a barcode reader can scan only one item at a time [3]. RFID tags are very effective in being read through a variety of substances and conditions such as extreme temperature, soil, dust and dirt, fog, ice, paint, creased surfaces, and other visually and environmentally challenging conditions, where barcodes technologies would be useless [4].

RFID technology is likely to increase the visibility and accountability in the supply chain. It is useful for governments, manufacturers, retailers, and suppliers to efficiently collect, manage, distribute, and store information on inventory, business processes, and security. RFID technology helps to automate workflow, minimize inventory and prevent business interruption in the assembly process. In the long term, RFID technology has the potential to help retailers provide the right product at the right place at the right time thus maximising sales, profits and preventing theft. To be functional, an RFID technology must be integrated with the various information

systems along the supply chain management to provide a meaning to the data and to allow for the information exchange of companies (e.g., healthcare, retail) using the technology. Despite these potential benefits and clear advantages over bar coding, there are challenges or issues with implementing RFID-enabled system applications in the global supply chain. We address and examine some of the important issues relating to RFID implementation in SCM system.

The paper is structured as follows: section 2 illustrates challenges relating to RFID implementation in the SCM system. In section 3 a case study of pharmaceutical SCM system is used to demonstrate the application of the practicality relating to RFID implementation issues. Section 4 provides a broad discussion of the results and section 5 concludes the paper.

2 RFID Implementation Challenges in SCM System

There are key issues that present a host of challenges for the successful implementation of RFID technology in the supply chain management system. Some of the challenges relating to RFID implementation in SCM systems are as follows:

a) *RFID Tags Read Rate* - In a product distribution setting, RFID readers are not always able to read all the tagged cases on a pallet on a 100% basis [10]; this becomes one of the major obstacles to RFID deployment in the SCM system.
b) *Presence of Metal Objects and/or Liquid Containing Items* - SCM is an area of operations that normally teeming with metal, liquid and harsh environments. Interference from metal objects (e.g., pharmacy shelves) and other RFID tags that generate electromagnetic energy [7], disrupt the RFID signal and make it challenging for many businesses to tag and track with their RFID-enabled SCM system. The RFID tag is also affected by objects surrounding it especially metallic containers/objects [8].
c) *Privacy* – Privacy issues loom as one of the biggest concerns to the success of RFID implementation in a SCM system. Concerns have been raised about the right to privacy being compromised by the emerging technology. An intruder with unauthorized readers can intercept the communication between the tags (goods or products) and RFID readers, between readers and the back-end database system in the supply chain, and can access sensitive tag information such as product ID, name, supplier, manufacturer, and so on. Privacy advocates express concerns that placing RFID tags in common items or products may allow them to continue to be tracked once purchased by consumers. A serious concern for consumers is that once they have purchased items (e.g., sleeping pills from a retail pharmacy), they do not want themselves or the purchased items to be tracked after passing the checkout [4].
d) *Cost* – A recent survey shows that high cost remains the primary roadblock to greater RFID implementation in a SCM system [9]. The cost of supply chain infrastructure (RFID hardware - tags, and readers, IT system, network infrastructure, system integration, process redesign, organization change,

labor cost, and so on) is high. The cost of RFID-tags is far more than a barcodes system. Even though the cost of tagging is decreasing, it is still significant.

e) *Lack of Standards* - Lack of standards is a major obstacle for the deployment and support for widespread use of RFID system in supply chains. There is no consistent or common standard for the air interface for players in the SCM currently. Item-level tagging is also necessary for most of the supply chain processes where the payoff occurs. In the SCM system, goods and products are very often broken out of pallets and move in inter-pack and individual configurations. Without clear RFID standards and data ownership policies, investment in RFID has been a difficult task [8].

f) *Tag frequency and Serialization* - Frequency and serialization is also a significant issue. Supply chain players such as manufacturers are very concerned as to which tag frequencies to use and where. With the serialization issue they are concerned about what to include in a tag's serial number. . Some want the tag serial number to contain intelligence (e.g., product ID) information; others do not want the tag intelligence information, rather a random serial number.

g) *Return on Investment (ROI)* - one of the major barriers for adopting RFID technology in a supply chain is the managerial concern of quantifying the cost-benefit of ROI (Return on Investment). The slow rate of ROI seems to be one of the major economic reason supply chain players hesitate to deploy RFID technology [8].

h) *Security of communication channel* - Most of the security threats in SCM are attributed to the security of the communication channel between authentic RFID readers, and the tags through the air interface (i.e., wireless communication). A RFID tag reading occurs when a reader generates a radio frequency "interrogation" signal that communicates with the tag (e.g., a tagged container), triggering a response from the tag [2]. Unauthorized readers can trigger this response by revealing the product information such as the drug ID and it is subject to misuse by hackers and criminals. Further, with respect to Read/Write (reprogrammable) tags, unauthorized alteration of product data is possible in the SCM system.

i) *Database security issues* - Data security is a major issue for wireless due to the nature of the transmission mechanism (electromagnetic signals passing through the air) in a RFID-enabled SCM system. The security of the supply chain database repository from unauthorized users (e.g. hackers, and others) is a major issue in RFID-enabled SCM systems. The transmission of the collected product data from a RFID reader over an intranet/internet to a remote database is vulnerable to the interception, alteration or destruction of signals [2].

j) *IT infrastructure* - Modifying existing business processes is a daunting task for players (e.g., supplier, manufacturer) in the supply chain that usually entails changes in RFID technology investment strategies. RFID-enabled system implementations are also part of SCM IT infrastructure, which often necessitates significant work process. The lack of various components such as RFID tags, networked readers, RFID-based SCM system applications,

intranets (LANs and WANs) and back-end database servers may affect enhancements to facilitate the implementation process [7].

k) *Lack of confidence* - Many supply chain players (e.g., wholesalers) are reluctant to invest in a RFID technology not yet widely adopted.

l) *Job Loss* – The penetration of RFID into the SCM system eliminates or transfer jobs, it may have to be submitted to supply chain staff negotiation. Many players (e.g., supplier) in the supply chain predict that the use of RFID in the SCM system may affect jobs due to the nature of the SCM systems automation [8].

3 RFID in the Pharmaceutical Supply Chain – A Case Study

To address the challenges and issues mentioned in the section 2, we present a case study on a Pharmaceutical Supply Chain Management System (PSCMS). We also integrate RFID technology with a multi-layer architecture for PSCMS. We outline a RFID model for designing PSCMS, which can help pharmaceutical companies to overcome these challenges by providing accurate, real-time information on products as they move to the value chain and by automating related business processes. An application of the architecture is also described in the area of RFID-based PSCMS.

3.1 RFID Model for the PSCMS

The pharmaceutical industry relies upon the integrity of many forms of data throughout the process of drug trials, suppliers (chemical plants), manufacturers, wholesalers, and retail and/or hospital pharmacy. RFID-enabled PSCMS are used to track their pharmaceuticals (e.g., sleeping pills), to prevent counterfeiting and loss derived from theft during shipment.

The product flows of RFID-enabled PSCMS are shown in Figure 1. It consists of a RFID tag, a reader and SCM IT systems. Each unique patient tag can be passive, semi-passive or active [2]. Passive RFID tags are used for both reading/writing

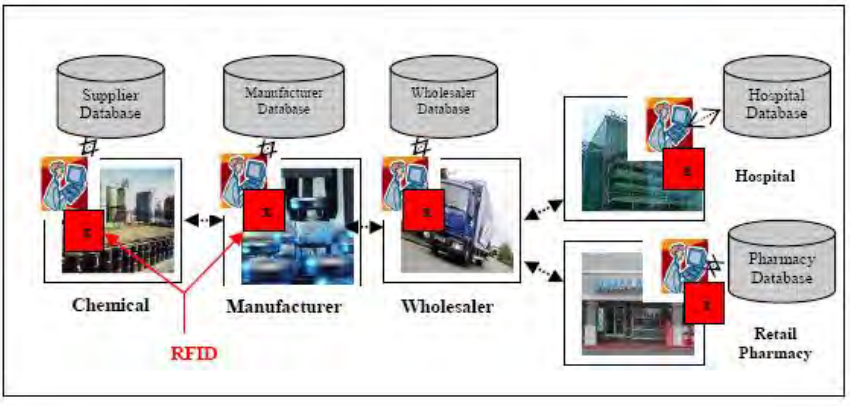


Fig. 1. Pharmaceutical Supply Chain System overview

capabilities by the reader. They do not need internal power (i.e., battery) as they become energized by the reader through radio waves and have a read range from 10mm to almost 10 meters.

PSCMS involves coordinating and integrating these flows both within and among pharmaceutical companies and a number of activities that are related to the movement of goods. The movement process includes placing or retrieving goods to and from the storage area or transferring goods directly from receiving to shipping docks [3]. RFID readers are placed in these receiving and shipping areas to read tagged items (e.g., containers in the chemical plants) for each supply chain player (e.g., manufacturer).

Chemical plants (i.e., suppliers) create raw materials and place them into containers or drums. A RFID passive tag is attached to each container to identify and serve as the data carrier. The passive RFID tag (with a high frequency of 13.56MHz) antenna picks up radio-waves or electromagnetic energy beamed at it from a RFID reader device (i.e., placed in the chemical plants shipping area) and enables the chip to transmit the container's unique ID and other information (if any) to the reader device, allowing the container to be remotely identified. The reader converts the radio waves reflected back from the tagged container into digital information which then passes onto SCM's IT system for processing. The system then stores the container's information into the supplier database before sending them to the manufacturer.

On receipt of the tagged containers, the manufacturer tracks the containers using RFID readers that are placed in the goods (e.g., raw materials) receiving area, verifies the quantity of goods being delivered and combines raw materials to make pharmaceuticals (e.g., sleeping pills). Pharmaceuticals are then place into tamper proof bottles and tagged with RFID tags. RFID readers (i.e., placed in manufacturer's shipping area) track tagged bottles and record the inventory into the manufacturer back-end database. The tagged bottles are then shipped to the wholesalers.

The wholesaler tracks and records each bottle that is shipped from the manufacturer. The pharmaceutical (e.g., sleeping pill) bottles are then shipped to the retail or hospital pharmacies.

Pharmacists or stores are equipped with RFID readers to verify that the tagged pharmaceuticals received from wholesalers originate from their purported manufacturer. RFID enabled PSCMS help pharmacists (both retail and hospital) to automate their existing systems, checkouts, and inventory and maintain their day-to-day activities very effectively and efficiently.

3.2 Multi-layer RFID Integration Architecture for PSCMS

Today most organizations, including the pharmaceutical industry, face data integration issues and RFID device management is a challenge while deploying RFID devices in their SCM system. Multi-layer RFID integration architecture establishes an infrastructure to address such a challenge, to automate and simplify the functionality for building RFID-based solutions in the SCM system. Figure 2 shows the integration layers (i.e., five layers) of RFID-based PSCMS architecture, namely, physical device integration layer, application integration layer, process integration layer, data integration layer and graphical user interface layers that make a complete integration.

The physical device integration layer consists of the actual RFID hardware components (such as RFID tag, and reader) that integrate with the existing SCM technologies for capturing the data automatically.

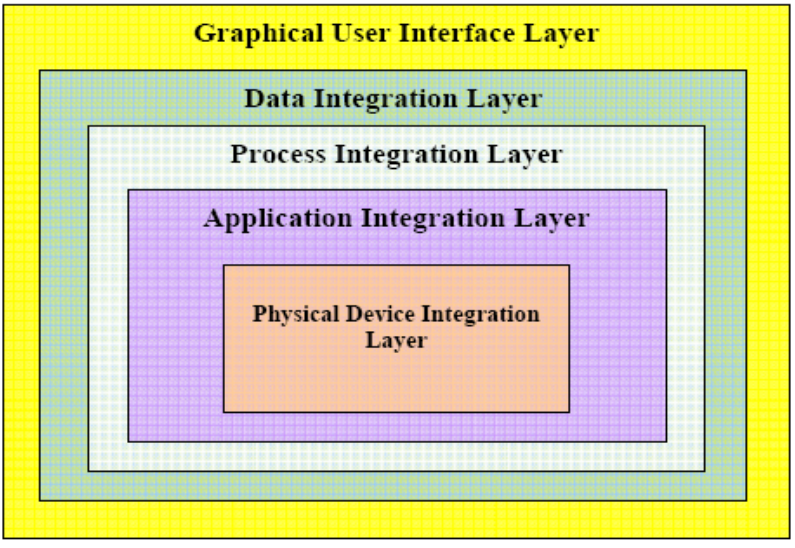


Fig. 2. Multi-layer RFID Integration Architecture for PSCMS

The application integration layer provides a middleware platform that integrates RFID systems with existing pharmaceutical applications, while providing adequate security. The middleware framework is an important element of RFID systems, which is viewed as the central nervous system from the SCM system perspective. It acts as the standard mechanism of communication with RFID readers. The middleware allows SCM computer information systems and applications to use a higher level communication methodology with the RFID infrastructure without having to understand the lower level issues required to communicate with the RFID hardware [5]. The application integration layer is responsible for monitoring physical device integration layer components and supports International Standardization Organization (ISO) standard [7].

The process integration layer provides a RFID business process development environment for defining supply chain business processes that provides real-time integration into their existing systems and automates the processes, thus optimizing the data flow. This layer enables data mapping, formatting, business rule execution and service interactions with databases in supply chain processes (e.g., wholesalers).

The data integration layer is composed of a RDMS (Relational Database Management System) and applications that allow supply chain managers to create RFID "events". This layer interacts with a SQL server and includes a data query/loading approach using SQL that supports high volumes of RFID data into a custom designed RFID database schema. These customized data (i.e., drug or product information) are then presented to the supply chain managers for fast and accurate pharmaceuticals (e.g., sleeping pills) identification [4].

Finally, the graphical user interface (GUI) layer is comprised of an extensible GUI (graphical user interface), which allows RFID devices (e.g., tags, readers) in a uniform, user-friendly way to work seamlessly in a Windows environment. This layer is

responsible for monitoring and managing the pharmaceuticals data and viewing the state of the system at each level. The GUI also helps in managing the information, generating various reports and analyzes the information at various stages in the entire value chain.

3.3 RFID-Enabled PSCMS Application

Figure 3 shows the RFID-enabled PSCMS application, which can be integrated with the pharmaceutical company's IT System for capturing goods and medication data automatically. The system is developed in Microsoft Visual Studio.net 2003 environment using Visual C++.net. The RFID-enabled PSCMS application issues a unique tag ID to every container (i.e., raw materials) or pharmaceuticals (e.g., sleeping pills) for each supply chain process (e.g., manufacturer) as shown in Figure 3.

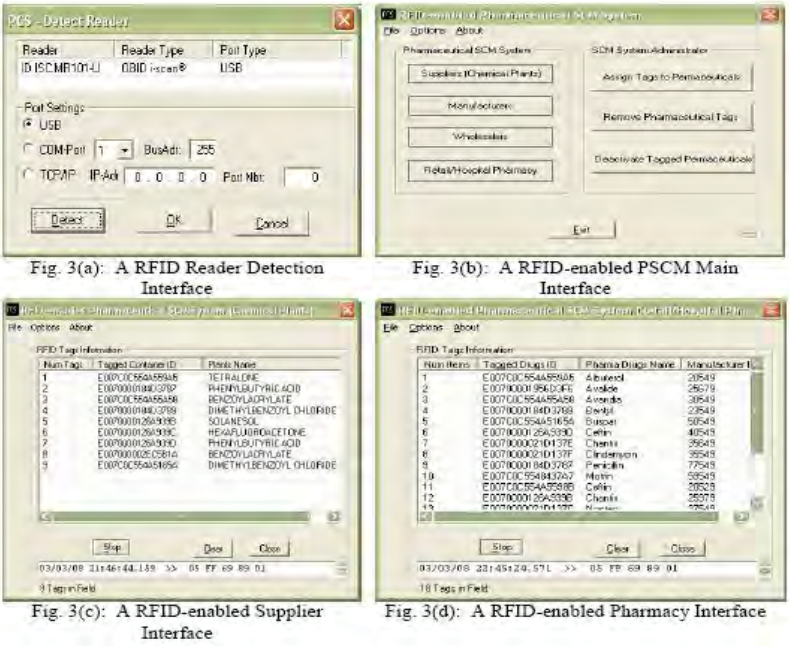


Fig. 3. RFID-enabled PSCM Application

The RFID then uses the tag ID as a key to information and perhaps other information (e.g. product name, manufacture date, price, and its distribution point) stored in the supply chain players back-end databases (i.e., SQL server). The tagged goods or product ID is used to identify each item all the way from suppliers, manufacturers, wholesalers, and retail and/or hospital pharmacy while in the supply chain. For example, an RFID pharmaceuticals tag only contains a unique drug ID, which a PSCMS application uses to retrieve a drug record stored in the supply chain players (e.g., hospital pharmacy) database. When a tagged drug appears within a reader (e.g., placed in hospital pharmacy) read range, the application (e.g., pharmacy interface) read the drug ID and associated drug information such as drug name, manufacturer, etc. automatically.

4 Discussions and Results

The case study in this paper demonstrates how to apply RFID implementation techniques in the pharmaceutical SCM system to examine three of those issues listed in section 2.

a) *RFID Tag Read Rate* - One of the primary reasons affecting the RFID tag read rate is poor positioning (i.e., tag misalignment), which results in a lack of sufficient energy. The other factors that affect an RFID tag read rate includes collisions on the air interface, tag frequencies, tag detuning, reader capabilities, operating environments (i.e., interference from other devices, temperature, humidity, and vibration), metal object and liquid containing items in the vicinity of the RFID-based SCM system. RFID tag and reader antennas can have a voltage interaction with products causing the tag to "retune" itself. It has been observed that the presence of metal and/water in the RFID tag vicinity causes a failed tag read within the SCM processes (e.g., chemical plants). As radio waves bounce off metal and are absorbed by water at UHF (ultra-high frequencies), it distorts the magnetic flux, thus weakening the energy coupling to the tag [5]. This makes metal containers difficult to tag and track with a RFID-based SCM system. In our RFID-enabled PSCMS, we have placed non-conductive material between the tag and the metal object to avoid scanning problems and improve tag reading rate. This technique is only useful if the size (e.g., one square feet) of metal object is small. However, in case of shipping containers, tags can be placed directly onto various types of extruded cylinders (surrounded by air space) by utilizing special mounting brackets. These mounting brackets are designed to drop down into the cylinder from above, eliminating the requirements to read tags from the side of the cylinder. We have also tested the use of foam balls to assist the RFID tag read rate in the pharmaceutical supply chain where tags are placed inside balls which, in turn, are placed in drums or cartons of products prior to shipment.

b) *Privacy and Security* - In a RFID enabled PSCMS, communication between RFID tags and readers, readers and back-ends database systems are one way. Our tags are passive, inexpensive and have a minimum amount of memory. We are keeping very little information in the RFID tag e.g., tag/product ID only and any private information is kept in a separate secure back-end database system (i.e., SQL server) to protect an individual's privacy. In our PSCMS, even if RFID tags hold information such as product ID, manufacture date, price, and its distribution point, the reader must be authenticated with a secret reference number (e.g., reader serial number) by the tagged items before reading them. Database modeling is an integral part of the supply chain player's database design process, which provides a systematic approach that supports the development of well-structured and high performance databases. A supply chain database design system is to validate and improve the quality of an existing design or even generate new designs. We also focus on taking advantage of an industry standard in the development of a supply chain database system solution that is flexible, robust, and meets the healthcare provider's requirements.

For the secure transfer of product data from RFID-enabled PSCMS to back-end database server, we are using a Hash Function-based Mutual Authentication Scheme [11]. This scheme, utilizing a hash function, is widely used for secure communication between readers and back-end SQL servers in a RFID-based environment.

In the worst case situation, if an intruder intercepts and gets the tag ID, he/she gains nothing because the tag does not contain any additional information.

c) *Standard* - The standard in a RFID system is critical for promoting companies in their supply chain development. RFID Standards are more likely to provide businesses with complete visibility of their supply chains, which often stretch across countries, industries and companies. There are number of well-established open standards such as the International Organization for Standardization (ISO) and EPCGlobal. As our RFID tags are ISO standard in PSCMS, the reader support and communicate with these (ISO 15693) tags effectively. Our proposed system uses passive tags (13.56 MHz ISO 15693 tag), which have a unique serial number with the read range of one meter. In the case of multiple product tag reading, if more than one tag answers a query sent by a RFID reader, it detects a collision. An anti-collision is performed to address this issue if multiple instances of tags are in an energizing field.

As the RFID technology becomes more widespread, the unit cost of tags and readers will continue to fall. Even though RFID tag operating frequencies and serialization differs between the USA, Europe and other regions, support for open standards such as ISO and multiple frequency readers make sense for global supply chains.

Experts predict that RFID will be a major advance in supply chain management, but supply chain players need to do considerable upfront planning and testing in implementing and/or integrating RFID technology successfully.

5 Conclusion and Future Work

In this paper we have outlined challenges relating to RFID implementation within the global supply chain. We have described a case study in the area of a pharmaceutical supply chain to address these issues. We have proposed a RFID model for designing a pharmaceutical supply chain management system to help pharmaceutical companies to meet/overcome challenges by providing accurate, automatic and real-time information on products as they move to the value chain. An application of the architecture is described in the area of RFID-based PSCMS. Efforts are being made to develop the complete system for use in supply chain management practice.

A better understanding of identified challenges relating to RFID implementation in supply chain will accelerate RFID technology introduction and further development of the technology itself. From there decision makers (e.g., supply chain managers) can move to developing RFID-enabled applications and integrating RFID data into existing applications.

The coordination of supply chain players seems to be a major factor for influencing the speed and ease of the RFID introduction process. As the sharing of open RFID system development and crucial information along the supply chain requires trust along the

supply chain standard procedures and regulations need to be in place. Despite the efforts of the two large standardisation bodies ISO and EPCGlobal, differences in regional standards remain a hindering factor in global RFID supply chain applications. Therefore, further research in RFID tag standards, the organisational and other issues, relating to the coordination and integration of supply chain members in regard to RFID applications and deployment should be conducted.

References

1. Fisher, M.L., Simchi-Levi, D.: Supply Chain Management (2001), accessed May 31, 2006, http://www.elecchina.com/en/doc/osy.htm
2. Stong-Michas,, Jennifer,: RFID and supply chain management: this amazing chip will change the world. Alaska Business Monthly, March 2006 issue (2006)
3. Rao, S.: Supply Chain Management: Strengthening the Weakest Link!, Team Leader for Industrial Automation, March 1, 2004 (2004), accessed on June 01, 2007, http://hosteddocs.ittoolbox.com/SR032604.pdf
4. Michael, K., McCathie, L.: The pros and cons of RFID in supply chain management. In: Proceedings of the International Conference on Mobile Business (ICMB 2005), July 11-13, 2005, pp. 623–629. Copyright IEEE (2005) ISBN - 0-7695-2367-6/05
5. Garfinkel, S., Rosenberg, B.: RFID Applications, Security, and Privacy. Addison-Wesley, New York (2006)
6. Chowdhury, B., Khosla, R.: RFID-based Real-time Hospital Patient Management System. In: Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science, and International Workshop on e-Activity 2007, July 11-13, 2007. IEEE Computer Society, Los Alamitos (2007)
7. Banks, J., Hanny, D., Pachano, M.A., Thompson, L.G.: RFID Applied, pp. 311–318. John Wiley & Son, Inc., Hoboken (2007)
8. Floerkemeier, C., Lampe, M.: Issues with RFID Usage in Ubiquitous Computing Applications. In: Proceedings of Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Springer, Berlin (2004)
9. Paul, R.A. (senior ed.): High cost slows RFID implementation (June 4, 2007), accessed on January 21, 2008, http://www.drugtopics.com/drugtopics/article/articleDetail.jsp?id=429850
10. Blair, P.: RFID Proving Ground Is All the World's Stage, METRO Group's (2007), accessed on January 24, 2008, http://www.rfidproductnews.com/issues/2007.09/metro.php
11. Lee, S.: Mutual Authentication of RFID System using Synchronized Information, M. Sc. Thesis, School of Engineering, Information and Communications University, South Korea (2005)

Design and Implementation of Clustering File Backup Server Using File Fingerprint

Ho Min Sung, Wan yeon Lee, Jin Kim, and Young Woong Ko

Dept. of Computer Engineering, Hallym University, Cuncheon, Gangwon-do, Korea
{chorogyi,wanlee,jinkim,yuko}@hallym.ac.kr

Summary. In this paper we proposes a clustering backup system that exploits file fingerprint mechanism for multi-level deduplication of redundant data. Our approach differs from the traditional file server system. First, we avoid the data redundancy by block-level fingerprint. The proposed approach enables the efficient use of the storage capacity and network bandwidth without the transmission of the duplicate data block. Second, we applied clustering technology because data transfer and I/O time is reduced a fraction of a percent for each node. In this paper, we made several experiments to verify performance of our proposed method. Experiments result shows that storage space is used efficiently and the performance is noticeably improved.

1 Introduction

Recently, there are growing needs to backup for most critical applications such as database, email, file server, web servers, and transaction servers. Backup is now accepted as the industry standard method for protecting important business and enterprise data. Current online backup technologies replace unreliable and time consuming tape backups with fully automated, reliable and scalable secure backups. However, online backup suffers from unlimited growth of protecting data.

To reduce backup storage capacity, the deduplication mechanism is widely used in the traditional backup system. The deduplication technology including file-level, block-level, and bit-level is used in backup system, however most of all backup system use only file-level deduplication because the overhead of deduplication algorithms(block-level, bit-level) is very high. The conventional file-level deduplication approach is based on file meta-data such as file size, the date of modification, file name and file hash information. Especially MD5 [20] and SHA1 [18] algorithm is well-known hash algorithm for file-level de-duplication. A block-level and bit-level de-duplication mechanism have a trade-offs; better de-duplication can be obtained by spending more processor and memory capacity.

There are enabling backup technologies which are improving duplicate elimination in storage systems including content-addressed storage(CAS) [3, 7, 8, 10, 13, 15, 23, 14]. The CAS provides a digital fingerprint for a stored data file. The fingerprint [17] ensures that it is the same exact piece of data that was saved. The CAS mechanism is based on file deduplication and ensures no duplicates files on the backup storage even when multiple copies of a file or attachment are being backed up. However, CAS also has limit on that it focus on file-level only deduplication.

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 61–73, 2008.
springerlink.com

To figure this problem out, we propose a cluster backup system that exploits fingerprint mechanism supporting block-level deduplication for removing redundant data blocks. Our approach differs from the traditional file server system. First we avoid the data redundancy by two-level fingerprint technology. The proposed system checks file duplication by file fingerprint mechanism and if there was no identical file fingerprint, then it tries to data block duplication check. It stores file data block if the data block fingerprint is not identical to existing data blocks, therefore it diminishes storage capacity extremely. The proposed system enables us to reduce storage capacity and network bandwidth efficiently without the transmission of the duplicate data block. Second we applied clustering technology to enhance data transfer capability and reduced I/O time a fraction of a percent for each node. In this paper, we made several experiments to verify performance of our proposed method. Experiments result shows that storage space is used efficiently and the performance is noticeably improved.

The rest of the paper is organized as follows. Section 2 surveys various well-known related works. Then in Section 3, we cover the design of cluster backup system and describe the mechanism of file finger print. In addition, we present implementation details and development environment. In Section 4, the result of experiment is thoroughly analyzed and then concludes in Section 5.

2 Related Works

There are lots of studies in eliminating data redundancy and fingerprint related research results [1, 2, 9, 11, 12, 21, 22, 16, 4, 19, 6].

Rsync [21] and ITSM [22] are a version management tools which support a redundancy elimination. Rsync is originally used for synchronizing files and directories from one location to another where connected by a slow communication link. The key feature of rsync is delta encoding based on the rolling checksum and multi-alternate search mechanism which reduce data transfer capacity and minimize data transfer time. To synchronize a file, it first splits a file into a series of non-overlapping fixed-sized blocks, and then produces 128-bit MD4 checksum for each block. If the checksum is identical to the opposite file block's one, the block is omitted to send, otherwise, the block must be transferred.

ITMS [22] is a centralized policy-based data warehouse which usually utilized in enterprise class data backup and recovery. The main function of ITSM is to insert objects via space management, backup, and archive tools, or to retrieve that data via similar recall, restore and retrieve methods. ITSM allows backup and restore of data "selectively", "incrementally" which is known as "Incremental Forever". Another key function of ITSM is a generating groupings of objects to be retained as a single unit which differs from traditional full/incremental style backup products in that the files are stored separately or in smaller aggregates rather than as a monolithic image.

Venti [16] is a block-level network storage system which is similar to the proposed system. Venti identifies data blocks by a hash of their contents, because of using a collision-resistant hash function (SHA1) [18] with a sufficiently large output, the data block can be used as the address for read and write operations. However, Venti is

intended for archival storage, it only supports block-level read and write. Therefore to build up full-featured backup system, file-level backup system is needed.

DHash [4, 19] is a distributed hash table working on Chord which is a scalable, robust distributed systems using peer-to-peer ideas. The key idea is to use erasure coding to store each block as a set of fragments, therefore it increases availability while saving storage and communication costs. Pastiche is a simple and inexpensive backup system which exploits excess disk capacity to perform peer-to-peer backup with no administrative costs. The underlying technology of Pastiche is a peer-to-peer routing, a content-based indexing and a convergent encryption. Pastiche is a distributed backup system therefore it can share CPU, disk capacity and network bandwidth from multiple nodes. However the network overhead produced by excessive routing requests leads to severe drawbacks for entire backup system. Furthermore frequently occurring node status changes (node join, node leave) make it difficult for the system to preserve consistency of backup data.

3 Cluster Backup System Design Issues

In designing the backup system, we considered two key approach, first is efficient storage management and the other is performance enhancement for large volume of data. For efficient storage management, we adapted aggressive data compaction using block-level deduplication mechanism. To provide performance enhancement, we used widely adapted clustering technology. With two technology, we can design and implement the online backup server which shows high performance and easy of use. The proposed system is composed of three main modules: client, MDS(Metadata server) and cluster file server. Figure 1 shows the architecture of the proposed system.

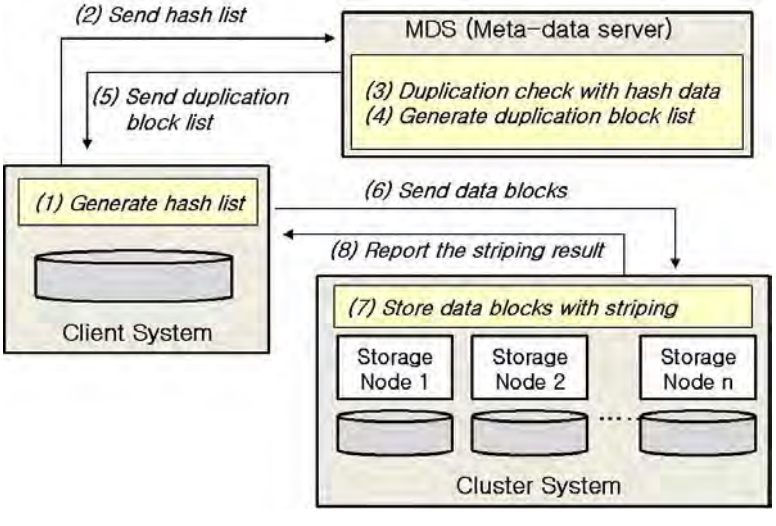


Fig. 1. The architecture of the proposed system

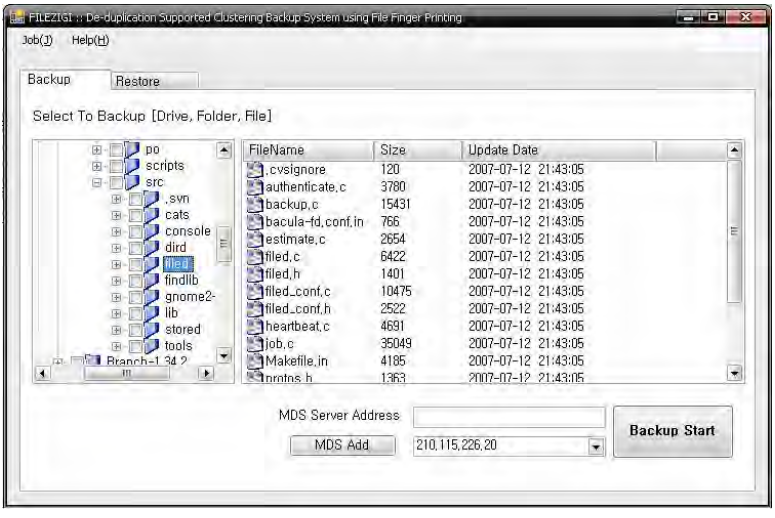


Fig. 2. Client program interface for the backup system

In the client module, the file is divided into fixed size blocks (about 16 Kbytes) and the each block is converted to hash data (about 160 bit). The hash data is grouped to hash list and then transmitted to MDS. The MDS has several roles. The MDS checks duplication blocks by examining overall hash list in hash database, and generates duplication block list. The result is transmitted to the client reversely. Finally, the client sends non-duplicating blocks to cluster server. However the cluster is composed of multiple storage nodes, before sending blocks, the client must decide to the node for each block. In our works, we used hash data as striping information to decide matching node, which leads to even distribution of blocks. The client module provides an interface for selecting backup files and directory for a user. If a user selects the file or directory, then the backup task is scheduled on a specified time or immediately. Figure 2 shows the client interface. The backup system supports multi-user capability, so the client can be performed simultaneously on each machine. The interface composed of two tabs(backup and restore). To start backup service, the user must specify backup directory and files. When backup start button is clicked, the backup procedure is on going.

3.1 Deduplication Mechanism

The key idea of deduplication is based on multi-level deduplication procedure. As mentioned on introduction section, traditional deduplication targets on file-level only, because block-level and bit-level deduplication leads huge amount of hashing overhead for checking duplicated blocks or a piece of data. To overcome those problems, we suggest the clustering approach which distributes hashing overhead over multiple nodes. The proposed algorithm divides file hash and block hash overhead to MDS and striping nodes. While a backup is processing, the client module creates a file list which includes backup files, directories and hash data. The hash data is generated by a SHA1 [18] hash

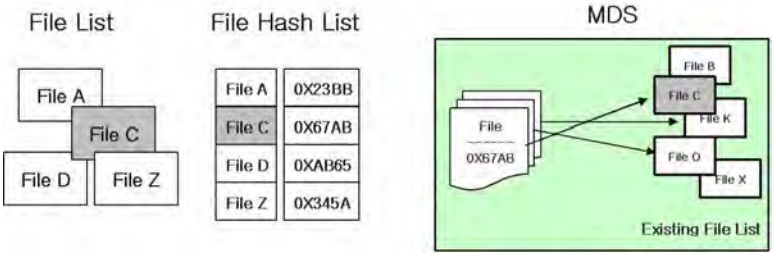


Fig. 3. Deduplication processing mechanism – file-level

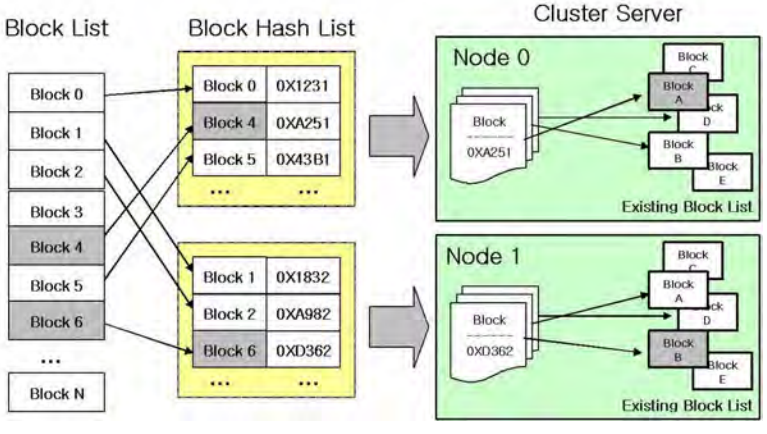


Fig. 4. Deduplication processing mechanism – block-level

algorithm that processes about 80 rounds arithmetic computation (add, and, or, xor, rotl, etc) for a one block data. The computation result of hash algorithm is a 160 bit binary data and stored with a plain hex code in the file list. Figure 3 shows the de-duplication steps in the proposed system.

In figure 3, with the file and hash list, we can find duplicated files in MDS, therefore we actually send backup files except File C. In this manner, a file-level de-duplication reduces the number of files to be backed up.

Into the bargain, we can reduce more data blocks by block-level de-duplication mechanism. For example, in figure 4, although File D has to be transferred to file server, not all blocks are transferred. The File D is composed of lots of blocks and there exists identical blocks with existing blocks in MDS. Following figure 5 shows a example of block hash results in the proposed system.

3.2 Main Algorithms for the Proposed System

In this section, we describe the key algorithms which generates block hash list and stripes data blocks.

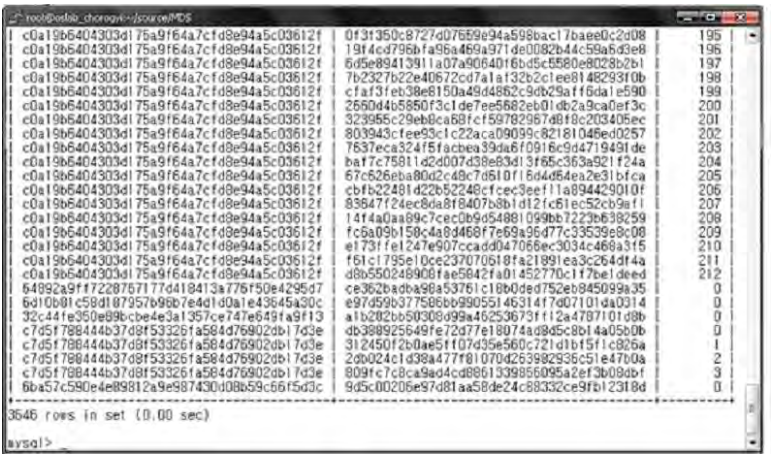


Fig. 5. The example of hash result

Algorithm 1. Generating Block HashList

Input. *FileStreamList*
Result. *HashList*
begin
  $Index \leftarrow 0$;
  for $FileIndex \leftarrow 0$ ***to*** $FileStreamList.Count$ **do**
    while $ReadBlock \leftarrow Read(FileStream[FileIndex], BlockSize)$ **do**
      $HashValue \leftarrow Sha1(ReadBlock)$;
      $LogSave(HashValue, Index, FileStream)$;
      $Index \leftarrow Index + 1$;
      if $HashValue \notin HashList$ **then**
        $HashList \leftarrow HashList \cup HashValue$;
      end
    end
  end
  return *HashList;*
end

The proposed algorithm creates fixed size blocks from file stream and then generates hash data with SHA1 hash algorithm. The algorithm continues to finish overall files to be backed up and finally returns block hash list.

If the client would like to send non-duplicating blocks to cluster server, before sending blocks, the client module must decide to the node for each block. In our algorithm, we used hash data and server list as a striping information to decide matching node. We simply distribute blocks to server nodes with a result of modular operation.

Algorithm 2. Block Striping Algorithm

```
Input. NodeList, HashList
begin
    Modulo ← 2^120 ÷ NodeList.Count;
    for Index ← 0 to HashList.Count do
        HashValue ← HashList[Index];
        NodeIndex ← HashValue ÷ Modulo;
        Node ← NodeList[NodeIndex];
        Node.HashList ← Node.HashList ∪ HashValue;
    end
end
```

3.3 File Backup and Restore Procedure

In this subsection, we discuss the procedure of file backup and restore in detail. We shows file backup procedure in two step. First is file-level deduplication processing procedure and the other is block-level deduplication procedure.

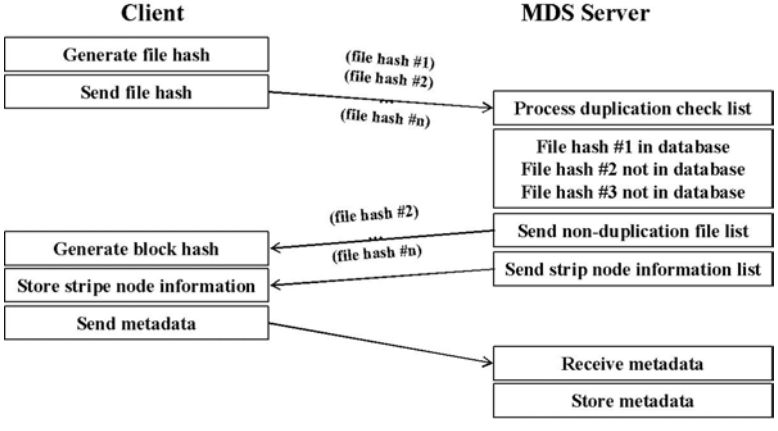


Fig. 6. File backup procedure(file-level deduplication processing)

Figure 6 shows a step in the file-level deduplication check and metadata management procedure. The client creates a file hash with the SHA1 hash function and then the file hash is delivered to the MDS. The MDS checks for duplicated files in the file hash database. After duplication checks, the MDS sends a non-duplication file list and stripping node information to the client. The client divides the non-duplication file to several fixed blocks and generates a block hash with algorithm 1. The client sends a block hash list and file metadata information to the MDS.

Figure 7 shows a block-level deduplication check procedure. The client generates block hash list for each cluster node with algorithm 2. And then the client send block hash list to each node for the checking of block-level deduplication. The cluster nodes

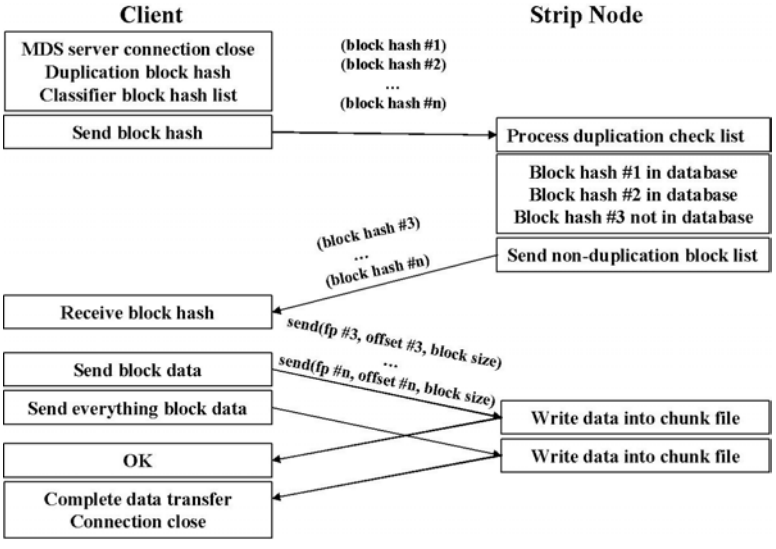


Fig. 7. File backup procedure(block-level deduplication processing)

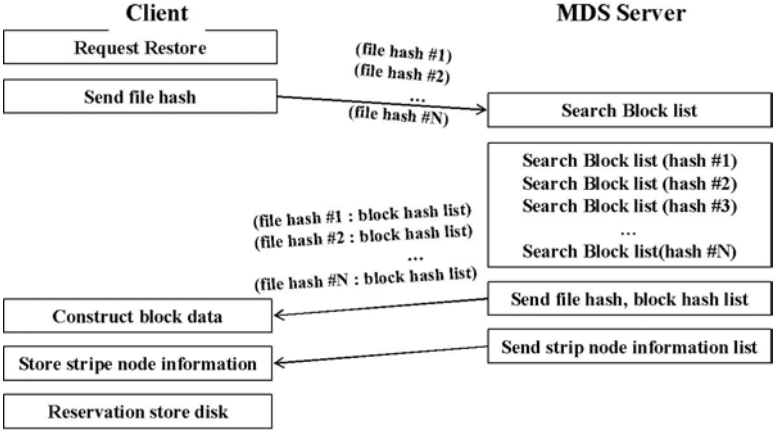


Fig. 8. File restore procedure 1

checks for duplicated blocks in local block hash database. After checking duplicated blocks, the cluster nodes send non-duplication block list to the client. The client sends only non-duplicated blocks to each node. If data block transfer is completed, then the cluster nodes writes each block and returns ACK to the client.

Figure 8 shows a restore procedure for backup data especially between the client and MDS. The client first composes file list for restoring from user request and sends file hash list to MDS. To restore target file, we needs the block hash list for the file, so MDS

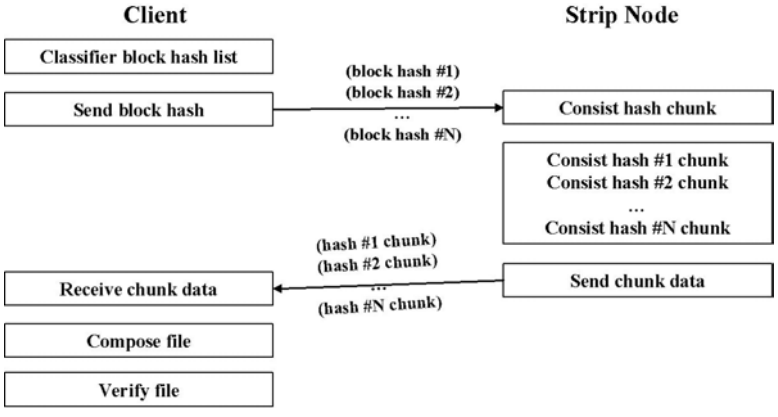


Fig. 9. File restore procedure 2

search block list with file hash data. For each file hash, we can retrieve block hash list and then the block hash list is returned to the client. The client divides the block hash list for each striping nodes, and then the client will send subset of block hash list to each striping nodes.

Figure 9 shows a restore procedure for backup data between the client and striping nodes. During the restore procedure in figure 8, the client completes to setup block list for each string nodes. And then the client sends the block hash list to the striping nodes. The striping node reads data blocks and then send it to the client. This procedure is processed on all striping nodes, so the client can receive large volume of data in parallel.

4 Experimental Result

All the performance data reported in this section were obtained on a 3GHz Pentium 4 desktop PC with 512Mbyte RAM, a WD-1600JS hard disk (a 7200RPM/8MB cache), and a 100Mbps Network.

4.1 Micro-benchmark

We first evaluated the overhead of each module in the client system. To acquire more accurate results, in this experiment, the measurement section was divided into five parts. The effect of the cluster architecture, from one node to four nodes, was also evaluated.

- File hash : file fingerprint creation time with SHA1 hash algorithm
- Block hash : block fingerprint creation time for non-duplicated files
- Metadata save : the measured time to save metadata

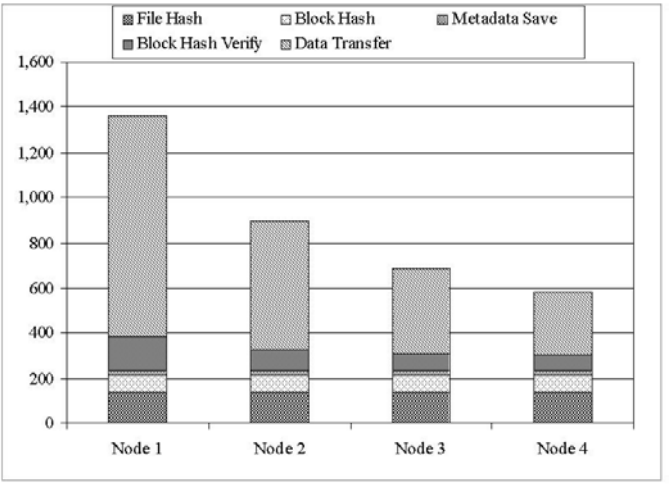


Fig. 10. Micro-benchmark timing (in sec)

- Block hash verify : the measured time for checking duplicated blocks
- Data transfer : data transfer time between the client and cluster nodes

Figure 10 shows the costs of each step in the proposed system. In this experiment, 1.5Gbyte files were used for backup, and a scalability of performance from 1 node to 4 nodes was measured. As shown in Figure 10, more cluster nodes give performance gains, especially in the data transfer and block hash verify sections, but the fingerprint-related section has no performance gains. As a rule of thumb, the costs related to block data transfer might be higher, and the file and block hash costs are also high. Whenever more nodes are added, the costs related to data transfer decreased sharply. Of course, this result shows only one side of the coin, but the result is meaningful in that an optimal number of cluster nodes can be found in the cluster backup system.

4.2 Macro-benchmark

Here, the compression result obtained for the processing of the backup data, the Linux distribution [5] and vmware [24] image, is presented.

In this experiment(Figure 11), the performance of the mail data backup (PST file of MS OUTLOOK) was evaluated, and the backup processing time was measured for the first version of the mail data (the size of the first-version file is 262 MB), and then the 294MB data were appended to the first-version file (the size of the second-version file is 556 MB). An additional 159MB data were also appended to the second-version file (the size of the third-version file is 715 MB). As shown in this experiment, all the data blocks were diminished, and the backup processing time was drastically reduced. The database backup experiment result is shown in Figure 12 A Mysql database file was dumped, and several versions of the file were made for testing purposes. The size of the first-version file is about 190 MB. Many SQL commands to the previous database were generated,

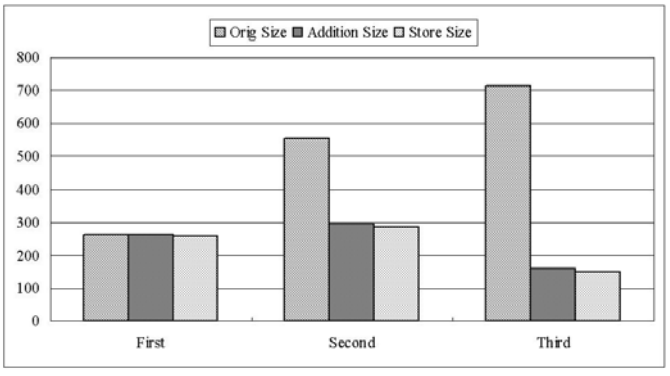


Fig. 11. Performance evaluation for mail backup (in Mbyte)

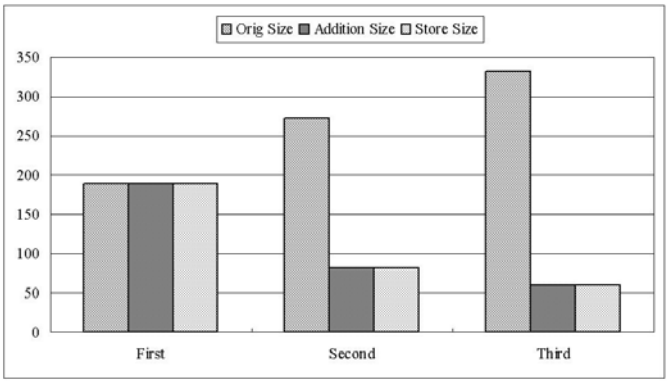


Fig. 12. Performance evaluation for Database backup (in Mbyte)

and second- and third-version database files were made, for the experiment. The sizes of the second-version and third-version files are 272 MB and 332 MB, respectively. As shown in the figure, the compression ratio drastically increased.

In this experiment(Figure 13), the performance of the Linux distribution and vmware [24] image was tested because this software is well known for its data duplication problem [14]. CentOS [5] was used as the Linux distribution test. The file size of CentOS is about 7.3 GByte. To produce a vmware image, Fedora Core version 3, version 4, and version 6 were installed, and the total image size was about 7.6 GBytes. Figure 13 shows the compression result of Linux distribution, with only the block-level deduplication file size reduced to 15%. Figure 13 also shows the compression result of the vmware image,and the compression ratio is about 15%. In this experiment, we did not adapt the file/block compression techniques, such as gzip and czip, as well as the performance tuning.

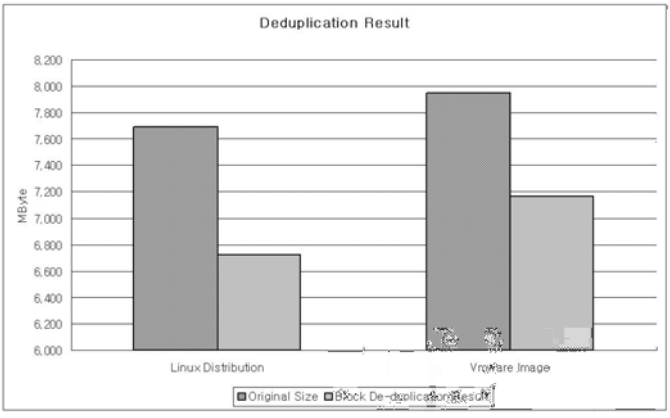


Fig. 13. Performance evaluation for Linux distribution and Vmware

5 Conclusion

The key idea of the proposed approach is to use multilevel deduplication for filtering duplicated files and data blocks. The proposed approach differs from the traditional deduplication-based file server system. The file redundancy was first eliminated by the file fingerprint, and the duplicated blocks were checked with a block fingerprint. Non-duplicated blocks are transferred to the cluster nodes with a simple block-stripping algorithm. The hash data are distributed between the metadata server and the cluster nodes; the metadata server checks only file-level duplication, and the cluster nodes check only block-level duplication. The proposed approach enables the efficient use of the storage capacity and network bandwidth, and the data transfer and I/O time are reduced to a fraction of a percentage for each node. Several experiments were conducted in this study to verify the performance of the proposed method. The experiment results show that the storage space is used efficiently and that the performance is noticeably improved.

Acknowledgement. This research was financially supported by Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

References

1. Ajtai, M., Burns, R., et al.: Compactly encoding unstructured inputs with differential compression. Journal of the Association for Computing Machinery (2002)
2. Annapureddy, S., Freedman, M.J., Mazires, D.: Shark: Scaling file servers via cooperative caching. In: 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation, Boston, MA (2005)

3. Policroniades, C., Pratt, I.: Alternatives for detecting redundancy in storage systems data. In: Proceedings of the annual conference on USENIX Annual Technical Conference, Berkeley, CA, USA (2004)
4. Cates, J.: Robust and Efficient Data Management for a Distributed Hash Table. Master's thesis, Massachusetts Institute of Technology (May 2003)
5. Centos homepage, http://www.centos.org/
6. Cox, L.P., Noble, B.D.: Pastiche: making backup cheap and easy. In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation (2002)
7. Bobbarjung, D.R., Jagannathan, S., Dubnicki, C.: Improving duplicate elimination in storage systems. In: Trans. Storage, New York, NY, USA (2006)
8. Forman, G., Eshghi, K., Chiocchetti, S.: Finding similar files in large document repositories. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, New York, NY, USA (2005)
9. Han, B., Keleher, P.: Implementation and performance evaluation of fuzzy file block matching. In: USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference (2007)
10. Kulkarni, P., Douglis, F., LaVoie, J., Tracey, J.M.: Redundancy elimination within large collections of files. In: Proceedings of the annual conference on USENIX Annual Technical Conference (2004)
11. Liben-Nowell, D., Balakrishnan, H., Karger, D.: Analysis of the Evolution of Peer-to-Peer Systems. In: ACM Conf. on Principles of Distributed Computing (PODC), Monterey, CA (July 2002)
12. Mogul, J.C., Chan, Y.M., Kelly, T.: Design, implementation, and evaluation of duplicate transfer detection in HTTP. In: Proceedings of the 1st Symposium on Networked Systems Design and Implementation (2004)
13. Nath, P., Kozuch, M.A., O'Hallaron, D.R., Harkes, J., Satyanarayanan, M., Tolia, N., Toups, M.: Design tradeoffs in applying content addressable storage to enterprise-scale systems based on virtual machines. In: Proceedings of the annual conference on USENIX 2006 Annual Technical Conference, Berkeley, CA, USA (2006)
14. Park, K., Ihm, S., Bowman, M., Pai, V.S.: Supporting Practical Content-Addressable Caching with CZIP Compression. In: Proceedings of the USENIX Annual Technical Conference, Santa Clara, CA (2007)
15. Policroniades, C., Pratt, I.: Alternatives for detecting redundancy in storage systems data. In: Proceedings of USENIX Annual Technical Conference (2004)
16. Quinlan, S., Dorward, S.: Venti: a new approach to archival storage. In: Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST) (2002)
17. Rabin, M.O.: Fingerprinting by random polynomials. Technical Report TR-15-81, Center for Research in Computing Technology. Harvard University (1981)
18. US Secure Hash Algorithm 1 (SHA-1). Request for Comments(RFC) 3174, http://www.faqs.org/rfcs/rfc3174.html
19. Rhea, S., Godfrey, B., Karp, B., Kubiatowicz, J., Ratnasamy, S., Shenker, S., Stoica, I., Yu, H.: OpenDHT: A public DHT service and its uses. In: SIGCOMM (2005)
20. Rivest, R.L.: The MD5 Message Digest Algorithm. Request for Comments(RFC) 1321, Internet Activities Board (1992)
21. rsync homepage, http://samba.anu.edu.au/rsync/
22. tivoli homepage, http://www.ibm.com/tivoli
23. Tolia, N., Kozuch, M., Satyanarayanan, M., Karp, B., Bressoud, T., Perrig, A.: Opportunistic use of content addressable storage for distributed file systems. In: Proc. of USENIX Technical Conference, pp. 127–140 (June 2003)
24. vmware homepage, http://www.vmware.com

Aspect Oriented Testing Frameworks for Embedded Software

Haeng Kon Kim

Department of Computer information & Communication Engineering,
Catholic Univ. of Daegu, Korea
hangkon@cu.ac.kr

Summary. Due to the complexity and size of embedded software together with strong demands on time-to-market and quality, testing is a crucial point that should be addressed during software development. Traditionally, testing is carried out during the last phases of the software development life cycle. As a consequence, testing activities are often subject to high time pressure, which either results in delayed market introduction or low product quality. The validation of functional and real-time requirements of embedded systems is a difficult task. It usually needs the electronic control unit and the controlled hardware components. But very often the hardware components are not available for testing the control software at the beginning of the development.

In this paper, we paper presents how test cases can be designed from use cases and how embedded control software can be validated without hardware components by simulating the test cases in early development phases using the AOP (Aspect Oriented Programming). For achieving an aspect oriented testable format, extended UML sequence diagrams are applied to formalize sequences of events, which have been specified in the use case scenarios. Provided that black box aspect oriented is used for developing embedded component applications, the monitoring of the dynamic behavior inside the components is not possible during simulation. But the simulated dynamic behavior is observable on the connections between the software components. In such a way monitored and recorded time stamp events are finally compared offline against the expected sequences of events specified in the test cases. The offline comparison validates the simulated behavior by demonstrating the fulfillment of user requirements and by detecting errors in case of contradictions during modeling.

Keywords: AOP(Aspect Oriented Programming), TDD(Testing Driven Development), Embedded Software Testing, Agile Modeling, Testing Frameworks.

1 Introduction

Embedded systems are control systems which are embedded in a technical system. They are designed for calculating actions as a response to characteristic input values. Usually this task is performed by a micro controller based electronic control unit which communicates with its environment by sensors and actuators. Distributed embedded systems consist of a network of hardware component units, which exchange information via a communication network [1, 2, 3].

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 75–88, 2008.
springerlink.com

Embedded software is a unique specialty within the broader software field. High-level IT systems generally run in clean environments and have little contact with the physical world. While bugs are always costly, bugs in PC software or large enterprise applications can be patched with relative ease. Effective methodologies for managing software risk and producing quality software are beginning to take root in industry. In embedded software development, testing and quality evaluation is one of the most important factors and can affect the entire embedded software development process. Embedded software systems are characterized by real-time requirements, distribution and increasing complexity. The validation of functional and real-time requirements of developed embedded software is a difficult task [1, 4]. The testing of embedded software is not available at the beginning of the software development, which leads to a late validation of the developed control software. The late validation results in a delayed detection of design errors in the control software, which causes increased fixing costs and delayed project schedules as shown in figure 1. These factors require lot of time, so reducing the testing and evaluating time is an effective factor to release the product early as shown in the figure.
In particular, among these practices, AOT (Aspect Oriented Testing) and Test Driven Development (TDD) stands out and prescribes that test code be programmed before the functional code those tests exercise is implemented.

Practicing TDD means designing software such that it can be tested at any time under automation. Designing for testability in TDD is a higher calling than designing good code because testable code is good code. Traditional testing strategies rarely impact the design of production code, are onerous for developers and testers, and often leave testing to the end of a project where budget and time constraints threaten thorough testing.

In this paper, we present here an integrated collection of concrete concepts and practices that are independent from platform-specific tools. In fact, our approach

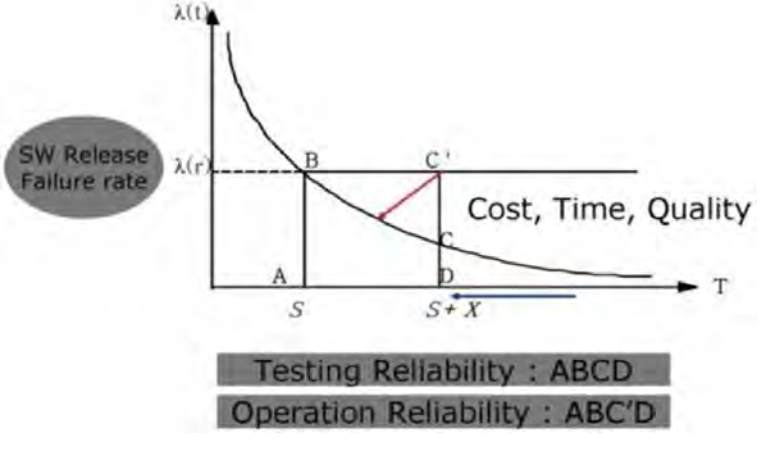


Fig. 1. Testing Reliability

drives the actual design of embedded software. we presents how test cases can be designed from use cases and how embedded control software can be validated without hardware components by simulating the test cases in early development phases using the AOP(Aspect Oriented Programming). For achieving an aspect oriented testable format, extended UML sequence diagrams are applied to formalize sequences of events, which have been specified in the use case scenarios. Provided that black box aspect components are used for developing embedded component applications, the monitoring of the dynamic behavior inside the components is not possible during simulation. But the simulated dynamic behavior is observable on the connections between the software components. In such a way monitored and recorded time stamp events are finally compared offline against the expected sequences of events specified in the test cases. The offline comparison validates the simulated behavior by demonstrating the fulfillment of user requirements and by detecting errors in case of contradictions. These strategies yield good design, systems that are testable under automation, and a significant reduction in software flaws.

2 Related Works

2.1 Embedded Software Testing Environment and Process

Testing in host environment; It is performed by using the emulator of embedded HW that is providing similar environment as target systems. It is very difficult to make precise emulation as real environment for the device, driver, memory capacity, CPU performance and network environment. It is usual unit and module testing rather than system testing in the host environment. It also focuses on the function and reliability than performance testing as time and resources. **Testing in target environment;** It is performed after develop the embedded software and decide the HW platform. Even the software is tested on host machine completely the Embedded software is usually developed on general development environment named as host environment (Windows, UNIX and Linux) and loaded on special embedded HW and OS named as target machine. It is also concurrently developed the HW and SW in many embedded systems.

Fig. 2. Typical Test Automation Process

Figure 2 shows the requirement and testing process for management and technical. Embedded software should be tested on target environment after the HW was decided. It has many constraints as memory and time resources, It is difficult to load the testing tool on the target machine. We need test tool in host machine and test agent on target machine side. General software products need 78% or more test coverage [5, 6]. Embedded software is very different because it has hardware dependent.

2.2 Testing Structures for Embedded Systems

Control software for embedded systems is characterized by real-time requirements, distribution and increasing complexity. The validation of functional and real-time requirements of developed embedded software is a difficult task. The testing of embedded software for example needs the controlled hardware components and special debugging devices for real-time data monitoring. But very often the hardware components are not available at the beginning of the software development, which leads to a late validation of the developed embedded software. The late validation results in a delayed detection of design errors in the embedded software, which causes increased fixing costs and delayed project schedules. The quality of prefabricated components is verified by extensive testing during their development. But when composing a new control software with such components the level of correctness can't automatically be considered to be equal to the original single components.

When applying such black box components for composing new control software, the monitoring of the dynamic behavior during simulation inside the components is supposed to be impossible. But the dynamic behavior of the component-based embedded software is observable on the connections between the connected components. For these purpose, especially the exchanged events between the connected components are monitored in the simulation model when executing the simulation of the component-based control software [1, 7].

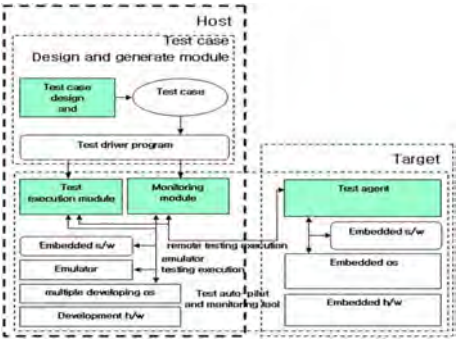


Fig. 3. Embedded Software Testing Structures

2.3 Aspect-Oriented Programming

Aspect-oriented (AP) is an approach to program development that makes it possible to modularize systemic properties of a program such as synchronization, error handling, security, persistence, resource sharing, distribution, memory management and replication [8, 9]. Rather than staying well localized within a class, these concerns tend to crosscut the system's class and module structure. An "aspect" is a special kind of module that implements one of these specific properties of a program. As that property varies, the effects "ripple" through the entire program automatically. Like object-oriented programming, AOP works by allowing the programmer to cleanly express certain structural properties of the program and then take advantage of that structure in powerful ways. In object-oriented programming, the structure is rooted in notions of hierarchies, inheritance and specialization. In AOP, the structure is rooted in notions of cross-cutting. As an example, an AOP program might define "the public methods of a given package" as a cross-cutting structure and then say that all of those methods should do a certain kind of error handling. This would be coded in a few lines of well-modularized code. AOP is an architectural approach because it provides a means of separating concerns that would otherwise affect a multitude of components that were constructed to separate a different, orthogonal set of concerns. AOP is the method for the improvement of the assembling process of software product line, the method that assembles core asset and variability is described by grammar elements such as joint point, point cut and advice without code-change. **Join points:** A well-defined point in the execution of a component. It can be a method call or execution, an access to an attribute, or the execution of a constructor. **Pointcuts:** A mechanism that encapsulates join points. It can be composed of one or more join points. **Advice:** specifies the action (i.e.: code) that must take place at a certain point cut (i.e.: a group of join points). With both abstractions mentioned above, advice gives developer the ability to implement crosscutting concerns. Intertype declaration mechanism allows developer to crosscut concerns in a static way. It permits alterations to classes and inheritance hierarchies from outside the original class definition.

Aspect is the container for the encapsulation of point cuts, advice code and an inter-type declaration. Acting like java classes, it can contain its own attributes and methods. Aspects are woven into classes to change class behavior and ultimately the behavior of the software product.

Concerns or aspects can be categorized into 2 types: core-level and system-level concerns. Core-level includes business logic and system level concerns include aspects that affect the entire system such as logging, authentication, persistency, performance. Many such system-level concerns tend to influence multiple implementation modules. They are called cross cutting concerns. Cross cutting concerns are features which cannot be otherwise be cleanly encapsulated in one development artifact and are tangled over several artifacts. Special composition rules combine the aspects with the artifacts with respect to reference points in the artifacts, the reference points are termed as join points. Separation of crosscutting features makes it possible to localize changes during maintenance,

customization and extension and thus improves productivity and quality. They affect many implementation modules even with programming approaches such as object-orientation they make the resultant system harder to understand design and implement. AOP focuses on identifying such crosscutting concerns in the system and implement them as a collection of loosely coupled aspect. AOSE (Aspect Oriented Software Engineering) employs abstractions known as aspects to separate these cross-cutting concerns throughout the software life cycle. AOP starts with a base component (or class) that cleanly encapsulates some application function in code, using methods and classes. One or more aspects (that are largely orthogonal if well designed) are applied to components, performing large-scale refinements that add or change methods, primarily as design features that modify or crosscut multiple base components. Aspects are implemented using an aspect language that makes insertions and modifications at defined join points in the base code at which insertions or modifications may occur. Join points may be as generic as constructs in the host programming language or as specific as application-specific event or code patterns. One such language AspectJ extends Java with statements such as "crosscut" to identify places in java source or event patterns. The statement "advice" then inserts new code or modifies existing code where ever it occurs in the program. AspectJ weaves aspect extensions into the base code, refining, modifying and extending a relatively complete program [10].

2.4 Aspect Oriented Testing and Test Driven Design

TDD is a style of programming where we write a test before the code that it is supposed to test [7, 8]. The test shouldn't even compile at first because the code to be tested hasn't been written yet. We then write enough code to get everything to compile, but purposely code it so that our test fails. We then write code to make the test pass. Each of these steps is critical to the TDD process. Finally, every time our tests pass, we refactor the code so that it is the best, resume-quality code that we can produce. We strive to make our code have the best possible design for the functionality we have implemented so far, but for nothing more than that functionality.

Only after we have refactored toward these goals do we go on to writing another test. We repeat this cycle of test-code-refactor in rapid succession throughout our development day.

TDD dictates that we work on a very small problem, solve it, and then move on to the next very small problem. After all, every large complex problem is just a bunch of little problems. We've found that by doing this, we can be successful throughout the entire day. Solving small problems, one after another, is sustainable, enjoyable, and highly productive. The key to these series of successes is rapid feedback. We run the compiler just as soon as we have enough code that stands a chance of compiling. Compiling after every single line may not be possible in practice, but it's not as far off as you might think. By writing our tests first, we guarantee that there are tests and that the system is designed to be testable [1, 3, 4].

2.5 Component-Based Embedded Systems

Embedded systems are embedded systems which are embedded in a technical system. They are designed for calculating actions as a response to characteristic input values. Usually this task is performed by a micro controller based electronic control unit which communicates with its environment by sensors and actuators. Distributed embedded systems consist of a network of unit, which exchange information via a communication network. A typical example for a distributed embedded system is a modern car. Nearly all kinds of control and supervision functionality, i.e. anti-blocking brake system, transmission control, fuel injection and different body electronics features, are controlled by distributed units. It can be presumed that almost the complete hardware is developed by using components off the shelf. In opposite to this fact it is a relatively new approach to design also the control software with predefined software components [11]. Development of component-based embedded software means that new application software for an embedded system is composed of a set of configurable (with parameters) software components, which have been explicitly developed for multiple usages. In the following, the term component always denotes a software component. Applying components and a component model promises many advantages as for example reduced development time and increased quality of the composed control software. The higher quality is reached by using prefabricated components, which have been extensively tested during their development. Applying uniform communication and execution mechanisms, provided by a component model, also reduces the number of possible design errors. The components exchange information by exchanging events and data on the connections between their interfaces [1, 4].

3 Aspect Oriented Embedded Software Testing Frameworks

3.1 Framework Architectures

Development of component-based embedded software means that new application software for an embedded system is composed of a set of configurable (with parameters) software components, which have been explicitly developed for multiple usages. In the following, the term component always denotes a software component. Applying components and a component model promises many advantages as for example reduced development time and increased quality of the composed control software. The higher quality is reached by using prefabricated components, which have been extensively tested during their development. Figure 4 presents our framework architecture for embedded software testing using aspect and TDD, which has been under developing with subset CASE tool of MoES(Mobile Embedded Software Development). MoES will support modeling, simulation and target code generation of mobile embedded software and has been applied in the context of this research work. MoES will offer different

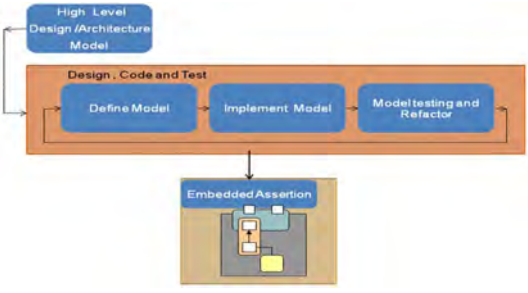


Fig. 4. Framework for embedded software testing using aspect and TDD

graphic and textual editors for modeling components and applications. The internal behavior of components can be specified in different notations, i.e. control block diagrams, state machine diagrams, a Java-based specification language or standard embedded code with specific platforms. TDD is primarily a design technique with a side effect of ensuring that your source code is thoroughly unit tested. However, there is more to testing than this. You'll still need to consider other testing techniques such as agile acceptance testing and investigative testing. Much of this testing can also be done early in your project if you choose to do so (and you should). In fact, in XP the acceptance tests for a user story are specified by the project stakeholder(s) either before or in parallel to the code being written, giving stakeholders the confidence that the system does in fact meet their requirements. With traditional testing a successful test finds one or more defects.

It is the same with TDD; when a test fails you have made progress because you now know that you need to resolve the problem. More importantly, you have a clear measure of success when the test no longer fails. TDD increases your confidence that your system actually meets the requirements defined for it, that your system actually works and therefore you can proceed with confidence.

As with traditional testing, the greater the risk profile of the system the more thorough your tests need to be. With both traditional testing and TDD you aren't striving for perfection, instead you are testing to the importance of the system. To paraphrase Agile Modeling (AM), you should "test with a purpose" and know why you are testing something and to what level it needs to be tested. An interesting side effect of TDD is that you achieve 100% coverage test — every single line of code is tested — something that traditional testing doesn't guarantee (although it does recommend it). In general I think it's fairly safe to say that although TDD is a specification technique, a valuable side effect is that it results in significantly better code testing than do traditional techniques.

3.2 Embedded AOP and TDD Cycle

Executing tests on the development system is both a risk and a risk reduction strategy. It is risk reduction, because we are proving the behavior of the code

prior to execution in the target. It is a risk because the target and the development platform compilers may have different capabilities, implementations, and bugs. Prior to target availability we risk using compiler features that are not supported by the target compiler.

To mitigate this risk we add another step to the embedded AOP and TTD cycle: periodically compile with the target's cross-compiler. This will tell us if we are marching down a path towards porting problems. The periodically mean code written without being compiled by the target's compiler is at risk of not compiling on the target. A target cross-compile should be done before any check in, and probably whenever you try out some language feature you have not used before. It would be a shame to use some platform dependent feature only to find that the target compiler does not support it. Once target hardware is available, our approach will continue to use the development systems as our first stop for testing. We get feedback more quickly and have a friendlier debug environment. But, testing in the development environment introduces another risk: execution may differ between platforms. To lessen this risk, we'll periodically run the unit tests in the frameworks prototype. This assures that the generated code for both platforms behaves the same.

Ideally the tests are run prior to check-in. We should consider how much of your work is being risked by not getting the feedback from running tests in the target. With target hardware available you should add tests for the hardware and/or tests for code that uses the hardware. Write tests that show how you expect to use the hardware and make sure it works as you expect.

The hardware engineers might just thank you. Automated unit tests are more difficult to create when the hardware interacts with the real world. The tests may involve external instrumentation or manual verification.

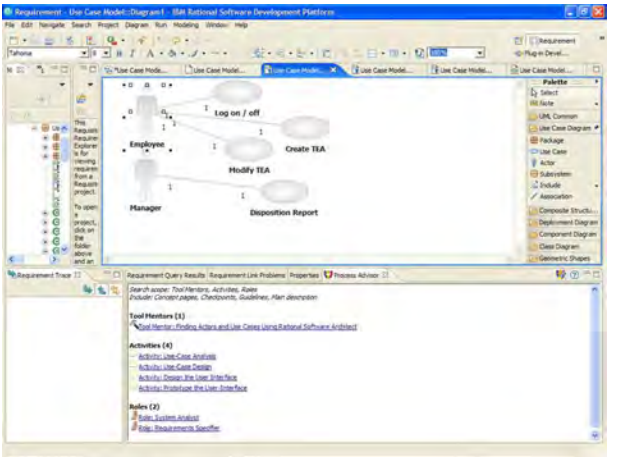


Fig. 5. Embedded AOP and TDD cycle

AOP and TDD frameworks validation process consists mainly of the following 7 steps as in figure 5:

1. Validate Use case analysis for gathering testable requirements from a user's point of view
2. Transformation of use case scenarios into extended UML sequence diagrams
3. Design and modeling of the embedded software and generation of a simulation model
4. Definition of dynamic sequences to complete the test cases
5. Simulation of the test cases for recording the communicated events between the embedded components in The simulation model
6. Offline comparison between the 'Simulated Event Sequences' (SES) and the 'Required Event Sequences'(RES) for validation and error detection
7. Correction of the embedded software or the use cases if a contradiction (possible error) has been detected.

TDD completely turns traditional development around. Instead of writing functional code first and then your testing code as an afterthought, if you write it at all, you instead write your test code before your functional code. Furthermore, you do so in very small steps — one test and a small bit of corresponding functional code at a time. A programmer taking a TDD approach refuses to write a new function until there is first a test that fails because that function isn't present. In fact, they refuse to add even a single line of code until a test exists for it. Once the test is in place they then do the work required to ensure that the test suite now passes (your new code may break several existing tests as well as the new one). Once your code works, you then refactor it to ensure that it's remains of high quality. This sounds simple in principle, but when you are first learning to take a TDD approach it proves require great discipline because it is easy to "slip" and write functional code without first writing a new test. One of the advantages of pair programming is that your pair helps you to stay on track. Since adopting TDD, we have almost never used a debugger on the code we have written. In fact, cranking up the debugger — or even adding a printf statement — is a sign of failure: we bit off a bigger problem than we could chew.

4 Modeling the Frameworks

In the context model of our work as in figure 6, a single test case consists of a stimuli sequence and an expected output sequence of events, which is denoted as 'Host Event Sequence (HES)'. The stimuli sequence defines a sequence of input events which trigger the simulation model. During simulation the monitored sequence of events is recorded in a 'Target Event Sequence' (TES). A test case is validated if the SES is conforming to the RES. A test case detects a possible error if a contradiction between SES and RES has been detected. In the following it is described how test cases are designed from use cases. A use case is a generalization of a usage situation where one or many actors interact with Test

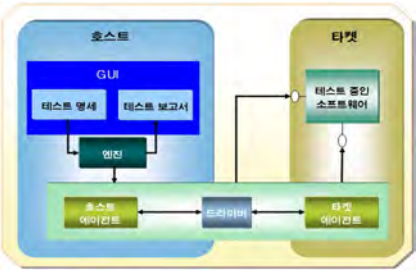


Fig. 6. Real environments for the AOP Testing

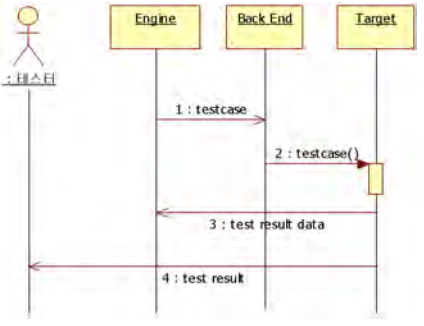


Fig. 7. The Flow of Back End

Fig. 8. Use case Model for the Frameworks

Fig. 9. Execution Example of the Framework

Case Design for the Validation of embedded Systems. One use case may cover several sequences of events — so called scenarios. A use case may be described either from an external (black-box) point of view or from an internal (white-box) point of view. An important advantage of use cases in general is their suitability for the software design as well as for the validation. The requirements do not come from the blue but are systematically gathered by writing use cases.

Use cases are not qualified to describe requirements completely, but when testing designed embedded software against all specified use cases by simulation later on, it can be stated at least that the validation is complete from a user's point of view. The figure 7 shows the flow of back end testing process and figure 8 show the use case model for our framework. Figure 9 show the example of execution the framework.

After the embedded model has been developed and a simulation model has been generated from it, the test cases can be completed. The expected output sequence of events is already defined in a SES. A stimulating input sequence of events has to be derived from the respective RES to complete a test case. The

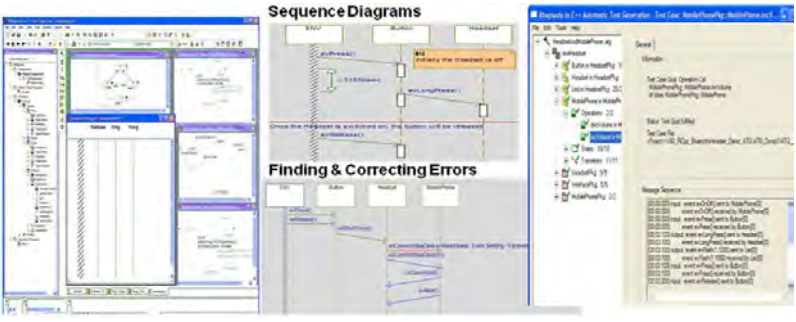


Fig. 10. Expected Output Sequences of Framework

first event of the TES usually becomes the first stimuli event. Alternatively other events which trigger this first event can become stimuli events as in figure 10.

5 Evaluation of Our Frameworks

There are several common myths and misconceptions which people have regarding AOP and TDD which we would like to clear up if possible.

Scalability

1. Embedded software with AOP and TDD test suite takes too long to run. This is a common problem with equally common solutions. First, separate the test suite into two components. One test suite contains the tests for the new functionality that you're currently working on; the other test suite contains all tests. You run the first test suite regularly, migrating older tests for mature portions of your production code to the overall test suite as appropriate. The overall test suite is run in the background, often on a separate machine(s), and/or at night. Second, throw some hardware at the problem.
2. Not all developers know how to test. That's often true, so get them some appropriate training and get those pairing with people with unit testing skills. Anybody who complains about this issue more often than not seems to be looking for an excuse not to adopt TDD.

Applicability

TDD, at the unit/developer test as well as at the customer test level, is only part of your overall testing efforts. At best it comprises your confirmatory testing efforts, you must also be concerned about investigative testing efforts which go beyond this.

Regression test suite

1. Although we can, and often do, create black-box tests which validate the interface of the component these tests won't completely validate the component, When you may have some reusable components and frameworks which you've downloaded or purchased which do not come with a test suite, nor perhaps even with source code.
2. The user interface is really hard to test. Although user interface testing tools do in fact exist, not everyone owns them and sometimes they are difficult to use. A common strategy is to not automate user interface testing but instead to hope that user testing efforts cover this important aspect of your system. Not an ideal approach, but still a common one.
3. Some developers on the team may not have adequate testing skills.

Tests form of design specification

The reality is that the unit test form a fair bit of the design specification, similarly acceptance tests form a fair bit of your requirements specification, but there's more to it than this. Agile developer do in fact model (and document for that matter), it's just that we're very smart about how we do it.

6 Summary

A significant advantage of AOP and TDD is that it enables you to take small steps when writing software. It is much easier to find, and then fix, those defects if you've written two new lines of code than two thousand. The implication is that the faster your compiler and regression test suite, the more attractive it is to proceed in smaller and smaller steps. Aspect oriented and test-driven design is a development technique where you must first write a test that fails before you write new functional code. TDD is being quickly adopted by agile software developers for development of application source code. TDD should be seen as complementary to aspect oriented approaches and the two can and should be used together. TDD does not replace traditional testing, instead it defines a proven way to ensure effective unit testing. A side effect of TDD is that the resulting tests are working examples for invoking the code, thereby providing a working specification for the code.

In this paper, we paper presents how test cases can be designed from use cases and how embedded control software can be validated without hardware components by simulating the test cases in early development phases using the AOP (Aspect Oriented Programming). For achieving an aspect oriented testable format, extended UML sequence diagrams are applied to formalize sequences of events, which have been specified in the use case scenarios. Provided that black box aspect oriented is used for developing embedded component applications, the monitoring of the dynamic behavior inside the components is not possible during simulation. But the simulated dynamic behavior is observable on the

connections between the software components. In such a way monitored and recorded time stamp events are finally compared offline against the expected sequences of events specified in the test cases. The offline comparison validates the simulated behavior by demonstrating the fulfillment of user requirements and by detecting errors in case of contradictions during modeling. In the future, we are going to develop the automatic tool based on this AOP and TDD that will have a big benefit for modeling and simulation of embedded software. It also enables the simulation of embedded control software for real-time behavior. Additionally it supports real-time data monitoring and recording of time stamp events, which is the necessary input for the offline comparison against the required sequences of events for the validation of the embedded software in early development phases.

References

1. Karlesky, M., Bereza, W., Erickson, C.: Effective Test Driven Development for Embedded Software. In: IEEE EIT 2006, East Lansing, Michigan (2006)
2. Van Schooenderwoert, N.: Mbedded Agile: A Case Study in Numbers (2006), http://www.ddj.com/dept/architect/193501924
3. Atomic Object: Test Driven Development in Embedded Software (2008), http://atomicobject.com/pages/Embedded+Software
4. Alles, M., Crosby, D., Erickson, C., et al.: Presenter First: Organizing Complex GUI Applications for Test-Driven Development. In: Agile 2006, Minneapolis, MN (2006)
5. Fleisch, W.: Simulation and Validation of Component-Based Automotive Control Software. In: Proceedings of 12th European Simulation Symposium, Hamburg, Germany, pp. 417–421 (2000)
6. Schmitt, W.: Automated Unit Testing of Embedded ARM Applications. Information Quarterly 3(4), 29 (2004)
7. Gunzert, M., Nagele, A.: Component-Based Development and Verification of Safety-Critical Software for a Break-by-Wire System with Synchronous Software Components. In: Proc. of Int. Symposium on Parallel and Distributed Systems Engineering (PDSE), Los Angeles, CA, USA, pp. 134–145 (1999)
8. Ruby Programming Language Ruby, A programmerś best friends (2008), http://www.ruby-lang.org/en/
9. Aspect Oriented Software Development: Aspect-Oriented Software Development Community & Conference (2008), http://aosd.net/
10. Eclipse Aspect project: Eclipse Aspect project (2008), http://eclipse.pse.org/aspectj
11. Atomic Object: Atomic Object (2008), http://atomicobject.com

Analysis on Malicious Peer's Behavior of the P2P Trust Resource Chain Model

Sinjae Lee, Shaojian Zhu, Yanggon Kim, and Juno Chang

Towson University
8000 York Road, Towson, Maryland 21252, USA,
Sangmyung University
Jongno-gu, Seoul 110-743, Korea
{slee5,szhu1,ykim}@towson.edu, jchang@smu.ac.kr

Summary. A malicious peer's behavior is one of the most challenging research areas of the P2P world. Our previous reputation-based trust model approach, which is based on the resource chain model (RCM), prevents malicious peers from spreading malicious contents among the open community. Moreover, a study on the malicious behavior over P2P community leads us to recognize a malicious node. The purpose of this paper is to identify malicious nodes based on the new resource chain model. In addition, the research includes the analysis of behavior of malicious peers on P2P networks. The study on various malicious behavior strengthens the existing resource chain model and allows us to keep P2P network much more reliable and stable.

1 Introduction

There are two main components, which are capability and behavior in the reputation system. Therefore, it is reasonable to analyze various peers' malicious behaviors. Moreover, recent studies have shown that there are many possible malicious peers' behaviors in the real P2P world.

There are numerous kinds of threat scenarios [3]. The first of its kind is malicious individuals or malicious peers that always provide inauthentic files [5]. The second kind is a malicious collective. Two malicious peers that know each other give each other good opinions and give other peers bad opinions. The third kind is a camouflaged collective. This is a tricky kind because a malicious peer sometimes provides authentic files to deceive good peers by giving them good opinions. The fourth kind is malicious spies are those peers of the collective that always give good files, but that always give good feedback to malicious peers.

Furthermore, numerous types of cheating are possible in the real P2P world [4]. A peer can exaggerate credibility. This means the peer intends to have a high credibility even though it actually has a low credibility. Another type of cheating is conspiracy. To put it precisely, a malicious peer can evade detection by using malicious neighbors. The third type is blame transfer. A malicious peer may try to blame a good node for hiding other malicious peers' misbehaviors. The fourth type of cheating is when a malicious peer can delete some information from other malicious peers' lists of interested peers by hiding consistency or by sending malicious information.

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 89–102, 2008.
springerlink.com

Numerous malicious actions can happen in the real P2P network [6]. First, attacking actions are possible. Eclipse [7] and resource-consuming attacks, distributing corrupt data, and attracting peers without serving them are representative of attacking actions. Abnormal behaviors, such as frequent joining/leaving and free riding, are also possible. According to "Resilient Trust Management for Web Service Integration [8]," the threat model presents two malicious behaviors - providing inauthentic web services and distributing dishonest opinions. On one hand, inauthentic web services can be harmful by propagating viruses and on the other hand, distributing dishonest opinion may cause a peer to choose less trustworthy providers.

Concerning of many possible cheatings [8], it is necessary to study behaviors of the malicious nodes. The resource chain model (RCM) is related to a simple malicious behavior [1, 2]. On one hand, this research aims to study the behaviors of malicious nodes and to determine the differences between the RCM and this research and on the other hand, the purpose of the analysis is to justify why it is necessary to use this paper.

It is indispensable to identify malicious peers because the original update information is not sufficient for the RCM to recognize that malicious information. More approaches on real-world P2P to increase the rate of successful downloads deserve careful attention. Although the RCM has been achieved some progress in network security, the limitations of the RCM oversimplify the activities of the malicious nodes.

It is appropriate to consider the malicious nodes doing various harmful activities in order to make the RCM more realistic and closer to the real world P2P network. The activities of the RCM's malicious nodes will always be opposite to those of normal nodes. For instance, although a malicious node does not have the resource, the node can pretend to have it. The importance of concentrating on malicious nodes' behaviors cannot be overemphasized in further improving the RCM security.

2 Identifying Malicious Nodes

The key point of our previous approach is that we try our best to find the best destination but it is not possible to guarantee that the candidate nodes are reliable. Consider a small scenario. All peers have their own unique identify. After the peers download resources and the transaction is successful, then the credibility of the nodes between the start and destination nodes is increased. However, if the transaction is unsuccessful, then the credibility of the nodes between the start and destination nodes is decreased.

Recent studies have shown that the resource chain, the best chain selection, the credibility update for the whole chain, and the neighbor list are maintained in the previous works [1, 2]. It is important to focus on collective historical transaction data and on the statistical analysis of data about the failure transaction table. It is also important to identify malicious node instantaneously. Having decided that the RCM has many advantages, the only thing left to discuss is its limitations due to the behaviors of malicious nodes. Recently, the RCM has attracted a considerable amount of attention as more realistic network model that closely approximates a real-world P2P network so it is important to identify and reveal such malicious nodes.

The limitations of the model due to malicious nodes need to be examined in detail. It is clear that malicious nodes will always do the opposite of normal nodes, as stated in

the previous papers [1, 2]. Consider this situation for example, even a malicious node does not have resources; however, the node pretends that it does. Such a node is simple to identify only if it always does the opposite of normal nodes; however, sometimes the node does the same thing as normal nodes. Therefore, it is very difficult to identify whether the node is malicious.

An important concept in the RCM is that data tables are stored in each node and will play a key role for data exchange. In the previous model [1, 2], the forwarding table information never changes so it is important to consider malicious node behaviors because they are common in a real-world P2P network.

Although the RCM is the local reputation model, the RCM can be considered as a part of a global reputation model, like that of broadcast or multicast models. Because it is a local reputation model, the RCM is a pure P2P model. As a result, this paper also has local reputation mechanisms.

Another factor can be introduced to identify malicious nodes. That factor is the honesty factor; H. H represents the probability that malicious nodes are telling the truth. At first, there is no difference among the H values of different malicious nodes. H is more like a statistical number whose value can be set at first, and, if necessary, gradually be updated as more data is analyzed. The thing behind this number is the assumption that malicious nodes will not always do the opposite; sometimes, they will hide deeper.

It is most important to consider the malicious nodes in this paper. One would not find useful data for malicious node analysis after a successful transaction. Although malicious nodes may have taken part in the transaction, they behaved positively; thus, no useful data is available for malicious data analysis. As a result, the RCM can use the data of failed transactions.

A failed transaction sometimes occurs. When it does besides the updating in the RCM, the credibility-updating process needs to inform the starting node (the node that sent out the initial request), so the node broadcasts or multicasts the transaction chain to everyone in the community or to whichever nodes have taken part in the transaction.

The importance of the failed transaction chain cannot be overemphasized. Peers broadcast or multicast chain information to everyone. In short, this is simplified chain information, which means that somewhere between starting node N0 and N33, for

Chain ID	Failure Route			
1	N0	N3	N12	N33
2	N1	N4	N5	
3	N7	N8	N9	N11
4	N5	N7	N21	

Fig. 1. Table of Failed Transaction Chain

Failure ID	Nodes List (Size varied)								
1	0	3	7	8	1	12	21	34	67
2	2	7	32	43	18	43			
3	4	3	12	54	16	55			
4	6	2	12	87	43	77	15		
5	7	9	12	87	23	92	23		

Fig. 2. Failure Table

example, there must be a malicious node. Consequently, all nodes in the RCM that get a table like the one shown in Figure 1 keep it for future analysis.

The method of identifying malicious nodes is called frequency identification. This could be the easiest and most efficient way to reduce the amount of risk in a P2P network. Those nodes that appear most frequently in peers' failure table would rarely be believed by other nodes Figure 1 shows a simple failure data table. Node N12 is possibly not reliable.

After some time, every node might have a table of failed transactions. For example, find the nodes that appear frequently in failure tables then compare their appearance frequency (AF) to H. If AF is greater than H, then those nodes are treated as malicious nodes. Here is a numerical example of this concept. Assuming that H = 0.5 and AF (12) = 4/5 = 0.8 $>$ 0.5, node 12 must be malicious (Figure 2).

The frequency could be misleading, however, as some of the nodes in between only act as forwarders, so it is essential to find more reliable method of discovering malicious nodes. Data mining is necessary for defining some ownsecure pattern or subroute, hence, partially preventing the frequency effect of good nodes.

Malicious nodes also send out this kind of failure transaction chain. They are actually totally faked. Because the chain is faked, the nodes, which are in the faked failure transaction chain, will know it is faked. Consequently, they directly know that that node is malicious. They will send out a correction message to warn others that the node is malicious. In this way, if a malicious node cheats and sends out faked information, it will help others identify it.

Malicious node N0 broadcasts out a failure table (Figure 3a). Thus, nodes N3, N4, N6, and N7 directly know that N0 is malicious because they did not take part in this transaction and because they can check their transaction histories. Good nodes broadcast back malicious node identification messages (Figure 3b). Figure 3b shows the message from N3. Thus, the left nodes, which do not know N0 is malicious at first, like N1. The node will get fake chains from N0, also malicious identification messages from N3, N4, N5, N6, and N7. Therefore, the node will believe several other nodes, not only one.

By analyzing the possible risks and threats in P2P networks, we can actually get the following three most popular threat models. Our improved resource chain model can be

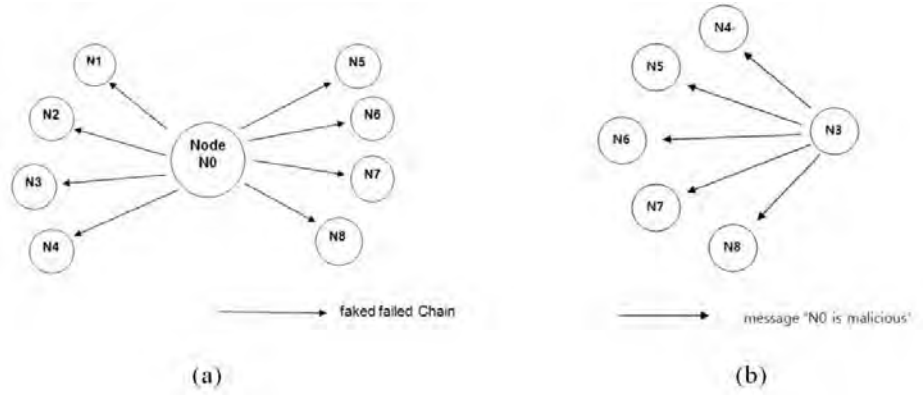


Fig. 3. (a) N0 Broadcasts Out Faked Failure Information (b) Good Nodes Broadcast Back

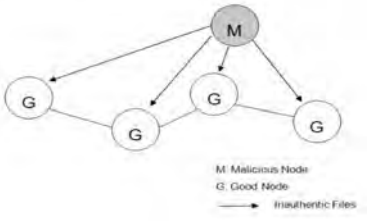


Fig. 4. Malicious Individuals

powerful enough to handle these possible risks to make our P2P network community secure.

The first type of threat model is actually called "malicious individuals" (Figure 4). The definition of "malicious individuals" comes from [3] and means a kind of peer that always provides inauthentic files. By applying this kind of threat to the RCM, we can say that in this kind of threat, a malicious node is always doing the opposite of normal nodes (those nodes that always provide authentic resources or information).

For example, given that there is a resource request by a node, which means the node is looking for the resource. Assume that another node gets this information and that it is a malicious node. As a result, although it has the resource, it will refuse the request. However, this action will not be easy because of the improved RCM.

In the improved RCM, a method called "forwarding requests" is used when a node does not have the resource. Specifically, assume that a malicious node has the request. The node will respond that it does not have the file, even if it does, and it will return the request by using "forwarding requests." This means that he will do exactly the opposite of normal nodes. In the first phase, whenever a transaction is finished, it is necessary to update the credibility based on the result of that transaction. This means that if it has a successful download, the RCM will give the node high credibility.

It is essential to update peers' neighbor lists depending on the credibility differences [1, 2]. Thus, malicious nodes will be given a lower credibility by nodes directly or

indirectly connected with them. Given that more transactions happen, the malicious nodes become increasingly isolated from the "normal node group" because other nodes update the malicious nodes' credibility time after time. This method is helpful for maintaining the system's credibility. By dynamically updating credibility, the method slowly removes malicious nodes from our interacting group.

If we say that the method of dynamically updating is somehow a passive or an indirect method of prevent the first threat, only after transactions, we would try to do the updating and the result of this updating would help prevent this kind of threat to some degree. However, by doing that, we can only get relationships between malicious nodes; we can never truly say we identify the actual malicious nodes. Identifying the actual malicious nodes is more powerful than identifying the relationships between malicious nodes, and it helps to directly increase the security of the network system.

In this paper, another method can be used to identify the malicious nodes. The method is called "Failure Transaction Collecting and Analyzing." It is based on the data collected from failed transactions. The method works like this: whenever there is a failure, the node acknowledges the failure and multicasts the failure data to its neighbors and to its neighbors' neighbors. Each node holds a kind of failure transaction table.

Whenever a transaction fails, assume that there is/are some kind of malicious node(s) in the route between the starting node and the destination node. For the first threat, the malicious node(s) will always does/do the opposite of the other nodes, thus it is necessary to look at the failure data collected. It is possible to get a list with each node's failure count by counting the number of times that a certain node is listed in the failure table. Malicious nodes have large failure counts.

Assume that the network has about 10% malicious nodes. The malicious nodes can be identified by picking the worst 10% of the nodes listed in the sorted failure count table. Peers will never connect with such nodes again. After getting the lists of the worst 10% nodes, it is necessary for peers to see the neighbor lists and to monitor all the possible transactions. These kinds of malicious nodes always perform these tasks poorly; thus, their behaviors will result in a bad reputation. The best way to confirm such nodes as malicious nodes is to check their credibility in the neighbor lists or through transactions.

The second type of known threat is camouflaged nodes (Figure 5) [3]. This kind of malicious nodes is smarter than the first kind of nodes, so they are actually more

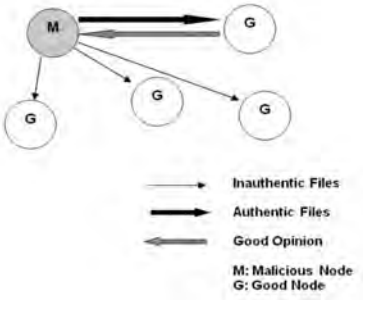


Fig. 5. Camouflaged Nodes

difficult to recognize. The reason that camouflaged nodes are smarter is that these kinds of nodes can partially provide authentic files to get good opinions; these kinds of nodes are the same as malicious nodes. However, even this kind of security threat is difficult to prevent in real-world P2P networks.

First, the credibility updating strategy is fair to every node that has taken part in transactions for a certain camouflaged node. Given that transaction T[0] with node list N[0], N[5], N[x] has finished successfully, if N[x] is doing well, it is given a good reputation from both N[0] and N[5]. However, another transaction T[1], that has the transaction list N[5], N[0], N[x] was unsuccessful. In that case, N[x] was a camouflaged node. Thus N[x] is given bad credibility value from both N[0] and N[5].

After T[0] succeeded and T[1] failed, the credibility of N[x] for N[0] and N[5] remains almost the same considering the symmetry of updating either positively or negatively. Assume that N[x] is doing more good things than bad things in a particular node's opinion. That node still slightly increases its credibility rate for N[x].

Our updating strategy is useful because if nodes behave badly, they will have the danger of getting a low credibility rate and would possibly be isolated in the future on the basis of their bad reputation. If these kinds of camouflaged nodes could perform more good transactions, they would get the same reputation or an even better one than normal nodes on the basis of the transaction quantity and quality.

However, the second method "Failure Transaction Collecting and Analyzing" will also work here because it is possible to focus on the failed transactions, so no matter how smart these nodes are, whenever they do something badly or not, it is essential to log that event.

The difference between camouflaged nodes and malicious nodes is that the camouflaged nodes partially do things well. The camouflaged nodes can have a result in better reputation than "pure" malicious nodes who always give wrong information. It is necessary to introduce another factor called the trust factor. The trust factor is a statistical indicator that indicates the probability of the camouflaged node behaving in a truthful manner.

Normal nodes have a trust factor of 100%, while malicious nodes have a trust factor of 0% because they never send true information. Camouflaged nodes can have trust factors between 0 and 100% and are typically about 40%. In 60% of these cases, the nodes will do well, having 3 successful transactions for every two false ones. As a result, these nodes get a credibility promotion of 3-2=1 for every five transactions, so these nodes gain a better reputation as time goes on. It is essential to determine whether some nodes are camouflaged by assuming that their trust factor is 40% in the neighbor list.

The third type of threat is the malicious/camouflaged collective which is the trickiest of all threats (Figure 6). In this case, malicious and camouflaged nodes combine to form a group, and they give each other good opinions and occasionally give bad opinions or lie during transactions when interacting with other nodes.

The strategy for updating credibility would be the perfect first step in dealing with this kind of threat. It is important to focus on the "chain" of the nodes in the RCM so that the starting node can find a destination node to get a requested resource. Given that two or more nodes are in the same chain and that they are badly performing transactions,

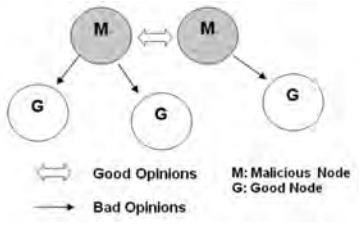


Fig. 6. Malicious/Camouflaged Collective

they both will get poor reputations; and no matter how much they increase each other's reputations, they will be unable to regain their trustworthiness.

It is also necessary to analyze the failure table data in order to consider the possibility of malicious/camouflaged collectives. If a node is at the end of a failure table, that node will have the most impact because it is the final node. If it happens to be malicious or camouflaged, each node from their neighbor lists will also be placed in that failure table, which will give normal nodes a partially negative reputation in the future. In that case, nodes at the end of a failure table should be given more attention and given more severe penalties when the nodes are caught cheating. Similarly, those nodes that come before the end of a failure table are given comparatively less severe penalties.

The next node, which is two nodes away from the destination node, would be given less severe penalties. An easy way to make use of this changing "impact factor" is to use the distance of the node in question away from the starting node. When we do the simulation, we are quite likely to identify the malicious or camouflaged nodes.

3 The Enhanced Resource Chain Model

In the previous model [1, 2], malicious nodes always do the opposite of normal nodes. We can use the two advantages of the RCM, namely, dynamic resource-chain-credibility updating and dynamic neighborhood maintenance. In addition, the previous simulation [2] has data, which successfully supported the RCM. It is important to dynamically maintain a sufficiently credible neighbor list and to increase the total rate of successfully downloaded resources when the number of transactions is increased in order to enhance the security of the P2P community.

There are various kinds of potential risks and threats in P2P networks. The RCM has some limitations in identifying malicious nodes, and it has difficulty in preventing other threats. In the enhanced resource chain model, it is necessary to focus on the enhancement of security, which takes into consideration more information about possible risks and threats and the solutions to them. It is also possible to identify the malicious nodes in the P2P network by using the histories of peers' behaviors.

As described in the previous section, there are many kinds of malicious behaviors. The most popular ones are the following three kinds of behaviors:

- Case 1: Malicious Nodes
  These nodes are the same as the nodes in the RCM. The nodes do the opposite of normal nodes.

- Case 2: Camouflaged Nodes
  These nodes behave smarter than the malicious nodes by partially telling the truth to confuse other normal nodes.
- Case 3: Malicious/Camouflaged Collective
  These nodes are the most dangerous. They not only behave maliciously, but also give good feedback to each other and give bad feedback to normal nodes.

The RCM has partially solved the problem of malicious nodes by dynamically updating neighbor lists. Hence, the malicious nodes in the RCM are isolated. It is very important to focus on enhancing the RCM with respect to the Case 2 and Case 3 threat groups.

The RCM was focused on increasing network security by maintaining reliable neighbor lists and by dynamically updating peers. However, the RCM only finished the work of isolating the abnormal nodes from the reliable nodes. It was previously impossible to identify the abnormal nodes, which behave maliciously in the RCM. In the enhanced resource chain model, it is necessary to use a new feature called Failure History Analysis, which can overcome the limitation of the RCM. It is essential to define some new terms before implementing the enhanced RCM. These terms are used to show the characteristics of different groups of nodes and to represent the new functionalities of each group of nodes.

- Trust Factor (TF): To simplify the analysis and grouping of all the potential malicious nodes in the RCM, we must use a TF value for each node in P2P networks. The TF is a percentage between 0% and 100%. It indicates the probability that a certain node is going to tell the truth. It is important to categorize the different groups of nodes in the RCM on the basis of their TF.
  - Normal Nodes: TF=100%; Normal nodes always tell the truth.
  - Malicious Nodes: TF=0%; Malicious nodes never tell the truth.
  - Camouflaged Nodes: TF=x%; Camouflaged nodes have varied TFs. X is between 0 and 1, which means that the camouflaged nodes sometimes tell the truth.
  - Malicious/Camouflaged Collective; each node has a different TF
- Failure Table (FT): Assume that there are failed transactions in P2P networks. After the failure, there are updated credibility chains and updated neighbor lists. Assume that nodes record and multicast the failure chain data to neighbors and that each node in the RCM records the failure data into the FT for analysis, which identifies abnormal nodes. The FT is a table structure in each node, which records all the failure transactions. In the RCM, the data in the FT can be analyzed to identify abnormal nodes.
- Abnormal Node List (ANL): The ANL is similar to the neighbor list, which stores the identities of nodes. However, the ANL only stores the identities abnormal nodes that are possible future threats.
- Impact Factor (IF): The Impact Factor is a new value in the enhanced RCM that enhances the updating of the credibility chain. The IF is a relative indicator that depends on the position of each node in the RCM. If a node is close to the starting node, then that node has a high IF.

Failure ID	Failure Chain						
1	N4	N0	N1	N2	N6		
2	N7	N9	N0	N1	N2		
3	N8	N0	N1	N2	N14	N16	
4	N6	N7	N8	N14	N15	N25	N23
5	.....	.....	.....	.....	.....	.....	.....
6	.....	.....	.....	.....	.....	.....	.....

Fig. 7. Failure Table

Rank	ID	Occurrence
1	N1	17
2	N3	13
3	N0	9 (14-5)
4	N2	1 (6-5)
...	...	...

Fig. 8. Updated Occurrence Table

After a certain number of transactions, nodes will update their neighbor's credibility lists on basis of transaction results. Each node in the chain gets the successful response from node N, the starting node. Therefore, each node in the chain increases its credibility for its directly connected neighbor. The following formulas are used to update credibility in the enhanced RCM.

- F1: CRnew = CRold x (1 + AwardFactor): CRnew is the new credibility after updating and CRold is the old credibility. AwardFactor is the bonus factor that indicates how much we should increase the credibility. After we introduce the impact factor (IF), the formula will be replaced by this enhanced formula, which takes into account the location effect in the RCM.
- F2: CRnew = CRold x (1 + AwardFactor x FLocEff (LocationofChain)): FLocEff is a function for calculating the IF on the basis of the location of each node in the RCM.
- F3: CRnew = CRold x (1 - PenaltyFactor): PenaltyFactor is a weight that indicates how much the enhanced RCM should decrease the credibility of a node.
- F4: CRnew = CRold x (1 - PenaltyFactor x FLocEff(LocationofChain))

After a node has a failure table (Figure 7), the failure data in it can be used for identifying abnormal nodes. There are many steps in the identification process. The first step is to refine the abnormal candidate nodes' group. This begins with the frequency statistics of each node in the failure table.

The occurrence of each node in the FT is sorted, and an occurrence table is made for the following step. The second step is to focus on the failure patterns, which are chins that have failed multiple times and are listed in the FT. For example, the failed chain is: N0 -> N1 -> N2. This chain failed five times and is listed in the FT with an occurrence of 5-the highest ranked failed chain in the FT.

The third step is divided into two substeps. The first substep is to match the occurrence and the patterns. For example, the pattern, N0->N1->N2, has the most number of occurrences, and N1 has the top rank in the occurrence table. The rank of N1 is higher than that of N0 and that of N2. The second substep is to eliminate the number of occurrences in the occurrence table. Node N1 is the most probable abnormal node. Therefore, the number of occurrences of N0 is subtracted from the number of occurrences of patterns, 5, in the updated occurrence table (Figure 8).

4 Simulation and Data Analysis

In the enhanced resource chain model, it is essential to add the trust factor (TF) for each node in the enhanced resource chain model. Each node has its own TF, which is used to classify the nodes as normal, malicious, and camouflaged. There are two advantages in using this type of simulation.

The first advantage is a high percentage of identified "malicious nodes" and "malicious collective." All "malicious nodes" and "malicious collectives" always lie during transactions. It is necessary to identify "malicious nodes" and "malicious collectives" because such nodes have a greater chance of attacking the P2P network than have normal nodes or "camouflaged nodes." Therefore, if a high percentage of malicious nodes is identified, the P2P network is more reliable.

Figure 9 shows that the recognition percentage increases quicker at the middle of the curve than the slope of the secant, which increases slower than the middle of the curve after about 6000 transactions. However, the trend line of the graph increases constantly during the time of the transactions. It is enough to prove that the percentage of recognized malicious nodes and malicious collectives is sufficient to maintain the P2P network security. The vertical axes represent the percentage of recognition rates and the horizontal axes represent the number of transactions involved (Figure 9). In the simulation, the simulator has the defined numbers (NodeTotal =10000, TimetoLive=5, TransactionTotal=10000, MaxNeighborNum=8).

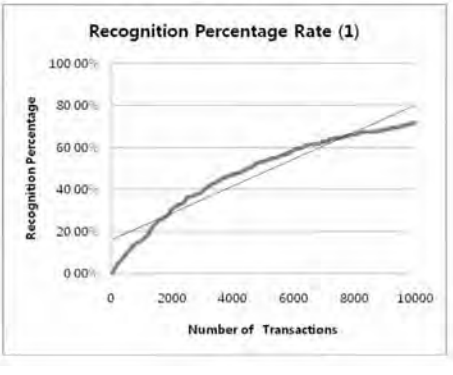


Fig. 9. Recognition Percentage Rate (1)

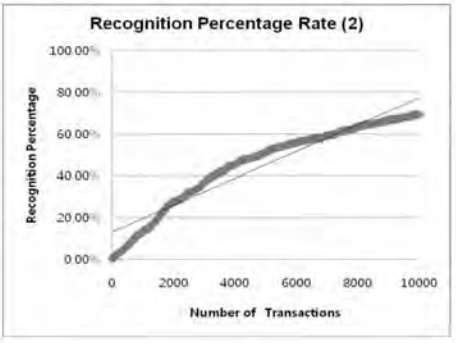


Fig. 10. Recognition Percentage Rate (2)

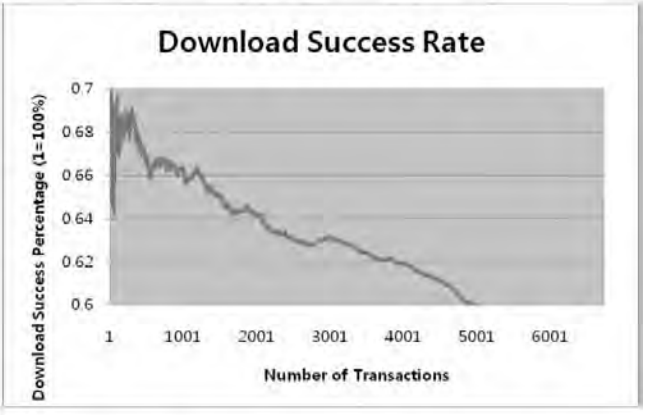


Fig. 11. Rate of Download Success without ERCM

The second advantage is a high percentage of identified nodes of all kinds of abnormal nodes with "camouflaged nodes." While "malicious nodes" always lie, "camouflaged nodes" sometimes tell the truth. In Figure 10, it is clear that the enhanced resource chain is powerful enough to increase the P2P network security. In Figure 10, the simulator results have the same scale as those shown in Figure 9.

The next simulation is to compare the rate of download success of the system using the enhanced resource chain model and that of the system without using the enhanced resource chain model. By using the same scale of the simulation results shown in Figure 9 and 10, it is possible to record the success rate of 10000 transactions. The diagram on Figure11 shows the change in the rate of download success of the system without using the enhanced resource chain model. The diagram of Figure 12 shows the change in the rate of download success of the system using the enhanced resource chain model. The results show that the rate of download success of the system using the enhanced

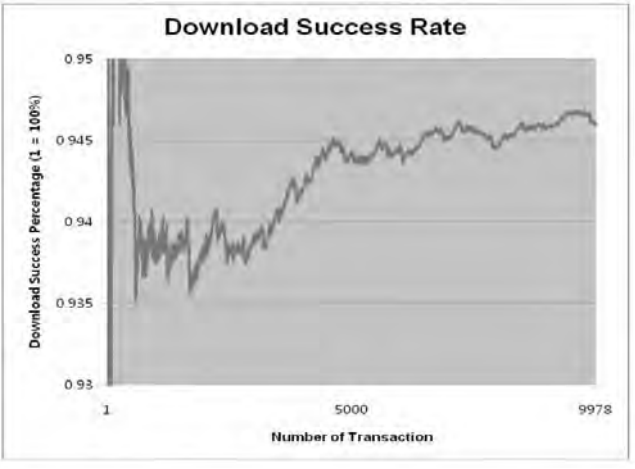


Fig. 12. Rate of Download Success with ERCM

resource chain model is much higher than that of the system without using the enhanced resource chain model.

The vertical axes represent the percentage of successful download rates, and the horizontal axes represent the number of transactions involved (Figure 11, 12).

5 Conclusion

There are two main components, which are capability and behavior in the reputation system. Therefore, it is reasonable to analyze various peers' malicious behaviors. Moreover, recent studies have shown that there are many possible malicious peers' behaviors in the real P2P world. While the previous model has global observations, this approach has local observations such as various malicious behaviors. It means must be concerned about the behaviors of a malicious node if we are to increase the security level in the resource chain model. In the previous work, the RCM assumes that malicious nodes never tell the truth. However, in this research, malicious nodes will partially tell the truth. The trust factor affects the percentage of their truthfulness. In addition, initially, if a download is unsuccessful, the whole resource chain is weakened. Conversely, in this paper, peers forward failure data to peers' neighbors' neighbors. Most importantly, through the data, it is possible to find ways to identify malicious nodes using the analysis of the failure table.

References

1. Lee, S., Zhou, S., Kim, Y.: P2P Trust Model: The Resource Chain Model. In: SNPD 2007, Qingdao, China (2007)
2. Lee, S., Kim, Y.: Analysis of the Resource Chain Model: the New Peer-to-Peer Reputation-Based Trust Model. In: CAINE 2007, San Francisco, USA (2007)

3. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: WWW 2003 (May 2003)
4. Ham, M., Agha, G.: ARA: A Robust Audit to Prevent Free-Riding in P2P Networks. In: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (2005)
5. Marti, S., Garcia-Molina, H.: Identity Crisis: Anonymity vs. Reputation in P2P Systems. In: Third International Conference on Peer-to-Peer Computing (P2P 2003), Sweden (2003)
6. Jin, X., Chan, S., Yiu, W., Xiong, Y., Zhang, Q.: Detecting malicious hosts in the presence of lying hosts in peer-to-peer streaming. In: Proceedings of IEEE International Conference on Multimedia Expo (ICME), Toronto, Canada, July 9-12, 2006, pp. 1537–1540 (2006)
7. Singh, A., Castro, M., Druschel, P., Rowstron, A.: Defending against Eclipse attacks on overay networks. In: Proc. SIGOPS EW 2004 (2004)
8. Park, S., Liu, L., Pu, C., Srivasta, M., Zhang, J.: Resilient Trust Management for Web Service Integration. In: Proceedings of the 3rd IEEE International Conference on Web Services, ICWS (2005)

Information Technology Service Management: A Thailand Perspective

Montri Lawkobkit

Dhurakij Pundit University
110/1-4 Prachachuen Road,
Laksi, Bangkok 10210, Thailand
montrilaw@gmail.com

Summary. This paper reports the results of a survey conducted at the first Thailand itSMF conference held in Bangkok, Thailand in November 2007. The results indicate that the adoption of Information Technology Service Management (ITSM) frameworks is driven mainly by multinational and large domestic companies and the ITSM frameworks are implemented mainly to improve the quality of service for the stakeholders as well as to comply with international standards. Information Technology Infrastructure Library (ITIL) appears to be the most popular framework for adoption. Training and certification emerge as key issues. Recommendations for further research conclude this article.

1 Introduction

Organizations today are increasingly dependent on effective and efficient Information Technology (IT) solutions to optimize business processes and to maintain and improve their competitive advantage. Implementing IT Service Management Framework Best Practices is proving to be an important approach for IT department in attempting to achieve this goal. At the same time, IT departments are under tremendous pressure to reduce costs, improve service levels and customer satisfaction, and deliver measurable business value. Such continuous improvement depends on a combination of people, processes and technology.

Information Technology Service Management (ITSM) is a subset of the Services Science discipline which provides IT operations delivery and support [11]. ITSM is not concerned with technical details of systems, but rather focuses on "providing a framework to structure IT-related activities and the interactions of IT technical personnel with business customers and users [9].The focus of ITSM is on the contributions IT can make to a business rather than on traditional technology-centred approaches to IT management. Because ITSM focuses on processes, it has much in common with process improvement approaches such as TQM, Six Sigma, Enterprise Resource Planning, and other similar frameworks and methodologies.

Information Technology Infrastructure Library (ITIL) service management framework has been developed by commercial organizations on the basis of the

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 103–111, 2008.
springerlink.com

original ITIL approach developed by the Central Computer and Telecommunications Agency (CCTA) in the UK in the late 1980s and probably the most comprehensive structured approach for providing IT services that is publicly available [4].The most common platforms emerging from or influenced by the original ITIL are Hewlett-Packard's IT Service Management Reference Model (HP-ITSM), IBM's Systems Management Solution Lifecycle (IBM-SMSL) and Microsoft's Operations Framework (MOF). The goal of any ITSM framework is to develop a set of best practices and guidelines concerning IT process [8].

However, adoption of ITSM frameworks is uneven and this may have implications for future expansion of business under Free Trade Agreements and for the ability of emerging economies to compete in global markets. Research indicates that approximately 60 per cent of European organizations have some form of ITIL implementation [7]. Adoption is particularly high in the Netherlands, parts of Scandinavia and the U.K. Generally, Europe is ahead of the United States and Asia Pacific region, where Australia is a leader in the region [7, 10]. It is estimated that 44 per cent of companies in Australia, 22 per cent in Singapore and 6 per cent in Hong Kong have ITIL frameworks, and that by 2010, ITIL will be in use in 30 per cent of companies with 250 to 999 employees and by 60 per cent of companies with more than 1,000 employees [5].

A range of ITIL-based frameworks have already been adopted by large organizations in the petroleum, financial and service industries. What the future holds for ITSM will depend on what we can learn from current practices. At the same time, an examination of current practices provides further insights into the relationships between ITSM and organizational factors.

The aims of this study are:

- to determine ITIL implementation success factors;
- to establish a reference benchmark for ITIL implementation progress and related IT framework processes in Thai organizations; and
- to delineate the perceptions of Thai IT professionals concerning the ITIL framework.

As a first step, a survey was conducted at the Thailand Chapter launch of the Information Technology Service Management Forum (itSMF) conference in Bangkok, Thailand in November 2007. The survey data help illustrate the current status of ITSM framework implementation; specifically, current ITSM frameworks implementation, planned ITSM frameworks implementation, reasons for implementation, and levels of satisfaction. These results present a snapshot of the status of the implementation of ITSM frameworks against which the IT community can benchmark its progress.

Three hypotheses were formulated:

- Hypothesis 1: Implementation of ITIL is positively associated with organizational size in terms of total IT users, IT budget and IT training budget;
- Hypothesis 2: Implementation progress of ITIL is associated with implementation of Service Support and Service Delivery processes; and

- Hypothesis 3: Satisfaction with the effectiveness of ITIL is associated with ITIL implementation progress.

2 Methodology

Each conference delegate was provided with a questionnaire at registration and requested to complete it at the conference. The questionnaire was distributed to over 200 IT professionals from over 40 organizations attending the conference. In total, 115 completed questionnaires were returned.

The questionnaire was adapted from the model reported in a similar study of ITIL adoption conducted by Cater-Steel and Tan (2005) at the itSMF conference in Australia in 2005 [2].The questionnaire was comprised of two parts. The first part contained items about the implementation of an IT framework within the respondent's organization. Part 2 asked about plans to implement a specific IT framework. The survey responses were anonymous.

3 Results

3.1 Status of ITSM in Thailand

The 115 respondents represent a cross section of IT professionals, ranging from technical specialists to presidents of IT departments. Nearly 40 per cent of the respondents were hands on managers and line staff.

Nearly half of the respondents (47%) said that their company has implemented an ITSM framework. These results are in line with surveys in other countries that show a trend towards increasing adoption of ITSM frameworks [3, 10] and indicate a high level of awareness among IT professionals. Of those companies which have implemented a framework, ITIL is the one most preferred by a large margin (21%). ITIL best practices has increased adoption because it serves as an open-platform ITSM framework and a de facto standard for implementing IT service management processes [1, 3, 10].

Thirty-nine per cent of the respondents reported that an IT framework was implemented mainly to improve the quality of service for stakeholders; to comply with management or business requirements and reduce costs; and to meet global company standards. This would seem to indicate a general shift in line with industry trends from purely technical service provision to the role of IT as a strategic function more directly related to the company's business objectives and to provide competitive advantage. The results also imply that corporate global policy has a strong influence on local IT implementation. The results indicate the same objective goal for implementing ITIL is to improve service quality [7].

Most of the respondents in this survey are employed by large domestic or multinational companies operating. Previous research shows that size and industry sectors are both factors in ITSM framework adoption, with larger firms having an advantage over smaller ones [6, 10, 12]. The results of this survey

support these findings. Most of the respondents in this survey act in advisory roles where they are involved in recommending and evaluating rather than key decision making.

The results also indicate that the size of a firm is a significant factor in terms of ITSM framework adoption. Thirty per cent of the firms implementing ITSM frameworks have large numbers of IT users. For a small country like Thailand with struggling small and medium enterprises (SMEs) and small business sectors, this has important implications for national productivity and competitiveness. Adoption of ITSM frameworks and best practices among smaller local companies and the public sector requires further research.

3.2 Satisfaction

More than half (60%) of the respondents reported they were satisfied with their IT framework implementation, although only a small number reported themselves as "Very Satisfied" (Table 1). This would indicate some room for improvement. Previous studies have identified a number of key success factors, including: commitment from senior management, the role of a champion to advocate and promote IT, the ability of IT staff to adapt to change, the quality of IT staff, and ITIL training [2].

Table 1. Satisfaction

Satisfaction	% Total (n=60)	
Very dissatisfied	1.67	1
Dissatisfied	1.67	1
Neutral	20.00	12
Satisfied	60.00	36
Very satisfied	16.67	10

The resources available to implement ITSM frameworks are another likely factor in success or failure, particularly the budget allocated to implementation. Over one third (38%) of the respondents reported that they did not really know how much was spent for the framework. Another 20 per cent would not disclose this information citing confidentiality. However, 62 per cent of the respondents said they spent more than 1 million Baht (approximately US$ 33,000) for their IT framework implementation. Whether this is above or below the industry average is difficult to say as there is no available data on this matter.

3.3 Training and Certification

Training emerged as a key factor in successful implementation of ITSM frameworks in the Thai survey. Almost 27 per cent of the IT professionals surveyed said that in-house ITIL training for their IT staff was necessary for implementation.

For an IT framework to work well and support the company's business goals, IT staff must know how to maximize the frameworks functions, and this is best achieved through training.

Most organizations agree that IT staff needs to be knowledgeable, and that training is one effective way of achieving this. However, the respondents' views regarding certification for existing and newly-hired IT staff were divided. Certification does not seem to be a priority among the companies surveyed. A little less than half (43%) of the respondents said their organization requires IT framework certification for current staff and 35 per cent said that certification is not a requirement. Companies appear to be less inclined to want certification for newly-hired IT personnel (35% required; 39% not required).

Thirty-one per cent of the respondents disclosed that their IT staff do not have certification while 19 per cent said that more than 10 of their staff have IT framework certification. Seventeen per cent on the one hand reported that 1 to 3 of their staff also has certification. A small percentage (3%) of the respondents said that 4 to 9 IT staff is certified to handle IT framework.

4 ITIL and Organization Factors

The data were entered into an Excel spreadsheet that was checked against the survey forms and converted to SPSS format to perform a number of simple statistical operations.

The results do not support Hypothesis 1. Implementation of ITIL is not positively associated with organizational size in terms of total IT users (Table 2). However, significant negative correlations were found between ITIL implementation and IT budget (Table 3), and ITIL implementation and IT training (Table 4). Therefore, there is support for the notion that IT budgets and IT training budgets decrease in size as ITIL implementation progresses.

Table 2. IT User and ITIL Implementation

	IT User	% Total (n=101)	
	<50	6.93	7
	51-100	13.86	14
	101-500	31.68	32
	501-1000	13.86	14
	>1000	33.66	34
Spearman's rho		ITIL	IT User
ITIL	Correlation Coefficient	1.000	-.068
	Sig. (1-tailed)	.	.250
	N	101	101
IT User	Correlation Coefficient	-.068	1.000
	Sig. (1-tailed)	.250	.
	N	101	101

Table 3. IT Budget and ITIL Implementation

	IT Budget (Baht)	% Total(n=21)	
	<300,000	23.81	5
	300,000-1,000,000	14.29	3
	>1,000,000	61.90	13
Spearman's rho		ITIL	IT Budget
ITIL	Correlation Coefficient	1.000	-.564(**)
	Sig. (1-tailed)	.	.004
	N	21	21
IT Budget	Correlation Coefficient	-.564(**)	1.000
	Sig. (1-tailed)	.004	.
	N	21	21

** Correlation is significant at the 0.01 level (1-tailed).

Table 4. IT Training Budget and ITIL Implementation

	IT Training budget (Baht)	%	Total (n=16)
	<300,000	62.5	10
	300,000-1,000,000	37.5	6
Spearman's rho		ITIL	Training Budget
ITIL	Correlation Coefficient	1.000	-.447(*)
	Sig. (1-tailed)	.	.041
	N	16	16
Training Budget	Correlation Coefficient	-.447(*)	1.000
	Sig. (1-tailed)	.041	.
	N	16	16

* Correlation is significant at the 0.05 level (1-tailed).

Hypothesis 2 postulated that implementation progress of ITIL is associated with implementation of Service Support and Service Delivery processes. When asked about current IT framework processes with ITIL implementation, 31 per cent of the respondents reported that they have implemented full process Service Support and 17 per cent have implemented Service Delivery (Table 5). This would seem to indicate that Service Support is, for some reason, a more significant factor in successful implementation than Service Delivery. It is common practice within many organizations to partially implement an ITIL framework rather than the full framework processes.

The data in Table 6 indicate ITIL implementation and satisfaction do not found provide support for Hypotheses 3 as the test resulted in a significant negative correlation. It is likely that factors such as commitment from senior management and implementation team play a role in the levels of satisfaction. IT departments were

Table 5. ITIL Implementation, and Service Support and Support Delivery

	Service Support	%	Total (n=70)
	Fully Implement	31.43	22
	Partially Implement	68.57	48
Spearman's rho		ITIL	Service Support
ITIL	Correlation Coefficient	1.000	.428(**)
	Sig. (1-tailed)	.	.000
	N	70	70
Service Support	Correlation Coefficient	.428(**)	1.000
	Sig. (1-tailed)	.000	.
	N	70	70
	Service Delivery	%	Total (n=46)
	Fully Implement	17.39	8
	Partially Implement	82.61	38
Spearman's rho		ITIL	Service Delivery
ITIL	Correlation Coefficient	1.000	.194
	Sig. (1-tailed)	.	.098
	N	46	46
Service Delivery	Correlation Coefficient	.194	1.000
	Sig. (1-tailed)	.098	.
	N	46	46

** Correlation is significant at the 0.01 level (1-tailed).

Table 6. ITIL Implementation and Satisfaction

Spearman's rho		ITIL	Satisfaction
ITIL	Correlation Coefficient	1.000	-.017
	Sig. (1-tailed)	.	.448
	N	60	60
Satisfaction	Correlation Coefficient	-.017	1.000
	Sig. (1-tailed)	.448	.
	N	60	60

forced to implement compliance with management or business' requirements and global company standards, but the same study would require further investigation.

5 Conclusions and Further Research

This survey indicates that implementation of ITSM frameworks in Thailand is at an early stage. Most implementations are taking place within large, multinational

companies in response to directives from corporate headquarters, large domestic firms with foreign partners, and smaller companies with a strong commitment to customer service. ITIL remains the most popular framework, with HP, IBM and Microsoft commercial vendors trailing rather far behind. Companies are generally satisfied with their implementations and rely heavily on training to ensure a successful implementation. Budgets allocated to ITSM frameworks implementation are modest and IT staff plays a largely advisory as opposed to decision making role in the process.

This survey was confined to a small sample of self-selecting firms and professionals who attended the itSMF conference, and provides an incomplete picture of ITSM. Further research is needed in this area. A more comprehensive and expanded survey would cover SMEs and public sector institutions, including educational institutions.

For Thailand's economy to continue to grow, domestic industry must become more competitive. ITSM frameworks can contribute to productivity and competitiveness, but only under the right conditions. The literature indicates what some of those conditions are, but not how they might apply in the Thai context. Further research is therefore required into domestic firms to determine what size and type of firms might benefit from adoption of ITSM frameworks.

Thailand's public sector remains largely unexplored in terms of IT framework adoption. Public sector institutions have demonstrated continuous improvement over the last few decades, but are nowhere near on par with the services offered by the public sector in other advanced Asian economies such as Singapore. Further research is therefore needed to determine the current status of and needed for ITSM frameworks in public sector.

Overall, the future for ITSM frameworks in Thailand looks promising. Much can be done to further promote the adoption of ITSM frameworks, especially if vendors and practitioners can work together under the umbrella of the itSMF.

Acknowledgement. The author wishes to thank the members of the Board of Directors of itSMF Thailand Chapter for their support.

References

1. Hochstein, A., Zarnekow, R., Brenner, W.: ITIL as Common Practice Reference Model for IT Service Management: Formal Assessment and Implications for Practice. In: Proceedings of the 2005 IEEE international Conference on E-Technology, E-Commerce and E-Service (Eee 2005) on E-Technology, E-Commerce and E-Service, March 29 - April 01, 2005, pp. 704–710. IEEE Computer Society, Washington (2005)
2. Cater-Steel, A., Tan, W.-G.: itSMF Australia 2005 Conference: Summary of ITIL Adoption Survey Responses. Technical Report. Toowoomba, Australia: University of Southern Queensland (2005)
3. Cater-Steel, A., Tan, W.-G.: Implementation of IT Infrastructure Library (ITIL) in Australia: Progress and Success Factors. In: IT Governance International Conference, Auckland, NZ (2005)

4. Rudd, C.: An Introductory Overview of ITIL Version 2. The UK Chapter of the IT Service Management Forum, itSMF Ltd (2004)
5. Holub, E.: Toolkit: Best Practices to Successfully Implementation ITIL. Gartner Publication Date: March 1, 2007, ID Number: G00146542 (2007)
6. Jin, K., Ray, P.: Business-Oriented Development Methodology for IT Service Management. In: Proceedings of the 41st Annual Hawaii international Conference on System Sciences (HICSS 2008), January 07 - 10, 2008, vol. 00 (2008)
7. Govekar, M.: Toolkit: European Polls on ITIL Adoption. Gartner Publication Date: April 11, 2007, ID Number: G00147236 (2007)
8. Salle, M.: IT Service Management and IT Governance: Review, comparative Analysis and their Impact on Utility Computing. HP Laps, Technical Report HPL-2004-98
9. OGC. IT Infrastructure Library - planning to implement service management. ITIL: managing IT services. Stationery Office, London. Office of Government Commerce (2002)
10. Bittinger, S.: ITIL Adoption in Asia Pacific - 2005 Update. Gartner Publication (2005)
11. Galup, S., Quan, J.J., Dattero, R., Conger, S.: Information technology service management: an emerging area for academic research and pedagogical development. In: Proceedings of the 2007 ACM SIGMIS CPR Conference on 2007 Computer Personnel Doctoral Consortium and Research Conference: the Global information Technology Workforce, St. Louis, Missouri, USA, April 19 - 21 (2007)
12. Tan, W.-G., Cater-Steel, A., Toleman, M.: Implementing centralised IT service management: drawing lessons from the public sector. In: ACIS 2007 18th Australasian Conference on Information Systems: the 3 Rs: Research, Relevance and Rigour - Coming of Age, Toowoomba, Australia, December 5-7 (2007)

A New Artificial Immune System for the Detection of Abnormal Behaviour

Rachid Elmeziane, Ilham Berrada, and Ismail Kassou

ENSIAS, Mohammed V University Souissi II, ALKHAWARIZMI Laboratory of computer sciences, BIRONI Team, BP 713, Rabat Morocco
{meziane,iberrada,kassou}@ensias.ma

Summary. We propose in this paper a new Artificial Immune System (AIS) named NK system, for the detection of abnormal behaviour with an unsupervised approach. Its originality resides in the unsupervised detection based on the mechanism of NK cell (Natural Killer cell) contrary to the existing AIS that use supervised approaches based on the mechanisms of the T and B cells. The NK cells develop the capacity to recognize the molecules of self-MHC through a unique class of receptors that can inhibit or activate its natural mechanism of the antigens elimination. In this paper, the NK system is applied to the detection of fraud in mobile phone. The experimental results are very satisfactory instead of the very weak proportion of the fraudulent operations in our sample.

1 Introduction

In order to solve complex problems of the real world in different areas such as optimization, detection of anomalies or robotics, heuristics inspired by natural processes have been used successfully. Indeed, in recent years, researchers have focused on the Biological Immune Systems (BIS), as an example of systems equipped mechanisms to allow and justifying a high degree of responsiveness and adaptability in protecting our self against attacks. Their work has contributed to the emergence of Artificial Immune Systems (AIS) as a new paradigm of artificial intelligence [8]. Although many industrial applications [7], scheduling [6], robotics [21] or intrusion detection [15] exist, few studies have addressed the problem of detecting fraud behaviour in telecommunications.

One of the most important roles of a biological immune system (BIS) is to protect the host against pathogens attacks, which proliferate in the body and cause damage to its proper functioning if they are not stopped and / or eliminated. The immune system is confronted with problems of detection, identification and response to pathogens. Several theories [3], [18], [23] have been proposed to explain how the immune system distinguishes elements (cells, tissues, substances, molecules) that belong to the human organism "Self" those harmful to the body "Non self". According to the theory of discrimination self/non self [3], [12], the surface of immune cells is equipped receivers capable of detecting non self exclusively, to bind with antigen and trigger mechanisms to eliminate it.

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 113–122, 2008.
springerlink.com

The theory of danger [23] provides a different view that the decision lies not with immune cells, but the cells of the body that the immune system is supposed to protect. According to the latter theory, the BIS are able to detect and respond to danger, rather than detect or respond to the Non Self. The cells assaulted are capable of guiding the immune response by producing danger signals which emit when they perish the abnormal behaviour [1]. These signals set up a perimeter around the danger assaulted cell, in which immune cells are able to identify the antigen. Furthermore, some viruses are able to avoid detection by their corresponding antigen and do occur after infected host cells. When this virus invades a cell of the body, it disturbs its operation [12]. The infected cells then express in their surface the different receptors those typically displayed [14], or do not show any more receivers [25]. Some specialized cells (NK cells (Natural Killer cells) and Tk) of the immune system are able to detect this change in the behaviour of infected cells. They are then activated and proceed with the disposal of infected cells.

In the problem of detecting fraud in telephony, a fraudster presents a priori legitimate behaviour (self), but it makes small fraudulent transactions (the so-called non-self) presenting a danger for the Telecom operator, insofar as it escapes the costs of these charges operation. In this paper we propose AIS for the detection of the behaviour of self/non-self with an unsupervised approach based on the mechanism of BIS said innate in particular the mechanism of NK cell. Unlike existing AIS which emphasize aspects of supervised learning and memory, adaptive BIS said adaptive mechanism including the T and B cell [11]. Such an approach allows, once the behaviour of self and non-self modelled, to distinguish them.

The existing AIS work as a supervised algorithm. They use a set of detectors, which characterize the abnormal instances of behaviour, by an evolutionary learning algorithm to determine the profiles of the anomalies in the sample [15]. In contrast, NK system works as an unsupervised algorithm. Our algorithm does not use the information concerning the characteristics of abnormal instances of behaviour. In analogy with the mechanism of NK cell, NK system characterizes and recognizes the patterns of instances of behaviour and then gives the clusters of these behaviours to be normal or abnormal.

This paper is divided into four sections. Section 1 recalls concepts and definitions of the mechanism of NK cells. Section 2 represents the NK system proposed based on the mechanism of NK cells for detecting abnormalities among a set of behaviours. In section 3, we apply the developed NK system to the problem of detecting fraud in telecommunications and we conclude in Section 4 and provide an overview of future work.

2 Mechanism of NK Cells: Concepts and Definitions

The innate immune system is the first line of defense of the human organism, especially the natural killer cells (NK cells) that can destroy cells infected by any antigen from the infection. These cells have no specific brands to the antigen as T

and B cells but recognize the complex molecules of self Major Histocompatibility (self-MHC). These specific allow the NK cells to recognize a self/non-self, or a non-self and destroy them with an unsupervised mechanism. Self-MHC molecules play an important role in the BIS, because they carry peptides (fragments of the protein chains) from the inner regions of a cell and present on its surface. This mechanism enables cells of BIS to detect infections within target cells without entering the cell membrane. There are several varieties of each self-MHC linked to the class of peptides. These molecules represent the patterns of the target cell.

The mechanism of NK cell can be summed up in four stages. (I) Firstly, the NK cell characterizes and recognizes the self-MHC molecule through its presence in the surface of the target cell. (II) Secondly, there is an initial reaction inhibitory through information reported by the reaction receptor / ligand. (III) Thirdly, there is another reaction receptor / ligand to activate the body's natural elimination of antigens. (IV) Finally, the two reactions produce two signals of a different nature which are compared to the activation or inhibition of the body's natural elimination of antigens [9].

3 Representation of Proposed NK System

We propose new AIS based on the mechanism of four phases of NK cells and can detect the presence of abnormalities among a set of behaviour. The NK system described below has four main phases concerning the recognition and extraction patterns instances (behaviour), and then transforming them into inhibition signal and activation signal to be ordered in a final phase to detect the presence of abnormal behaviour.

Phase 1: Recognition and extraction patterns of instances

- Representation of instance of behaviour in binary form.
- Representation patterns instances by permutations operations via GRP algorithm [20].

Phase 2: Modelling Signal inhibition sig_{kir}

- Generation signal inhibition by permutation entropy [2] of each consecutive and tangled block of patterns instances.

Phase 3: Modelling signal activation sig_{kar}

- Generation signal activation by the Damerau-Levenshtein distance [17] of each two consecutive patterns instances.

Phase 4: Detection of abnormal behaviour

- Analysis of each signal sig_{kir} and sig_{kar} by the MUSIC algorithm [22] to produce their spectral density PSD.
- Detection of the presence of an area of abnormal behaviour by differentiation of each PSD signal and then identifying these abnormalities by a finite impulse response (FIR) filter of two signal sig_{kir} and sig_{kar}.

Four types of operators are applied in these phases which depict the mechanism of NK cells and use in entry a time series. These operators will be described in the following paragraphs.

3.1 Description Operators of NK System

MHC operator

The first operator, the operator called MHC and noted op_{MHC}, modeled, in Phase 1, the characterization of self-MHC molecules by NK cells through the permutation operation GRP used in the encryption algorithms [20]. This operator allows recognition and retrieval patterns instances of a time series that we represented in binary form. Note that the self-MHC molecules are usually modeled in the AIS by permutations of variables positions [15]. The result of this operator is a series where each vector component is a permutation. Several permutation instructions such as PPERM, GRP, CROSS, OMFLIP and BFLY have been proposed for arbitrary bit-level permutations [20]. We adopt group of permutation GRP for its simplicity to implement.

Figure 1 shows a schematic of the MHC operator where X represents a binary instance before the implementation of the GRP algorithm, Y represents the position of variables associated with this instance and I is the instance after applying GRP algorithm. The values of Y are divided into two groups depending on whether the value of X is 0 or 1. Both groups are then placed in left I, if the value of X equals 0, and right I otherwise.

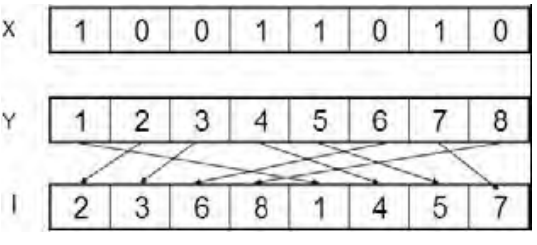


Fig. 1. Sample MHC operator

KIR-operator

The second operator, the operator called KIR and noted op_{kir}, modeled in Phase 2, the inhibition mechanism of NK cells using a permutation entropy, which on the basis of successive and tangled block of values of neighboring of a time series, measures the information from regular, random or chaotic behaviour [2]. This operator allows the transformation patterns of instances for inhibition and provides a signal noted sig_{kir} which we detail the calculation below.

The MHC operator models the characterization and recognition of NK cells of self-MHC molecules. The result is a time series of permutations which is

difficult to specify the party representing the intrinsic behaviour of self/non-self, if it exists. Complexity measures of information have been developed to compare the elements of a time series and distinguish the regular (eg periodic), chaotic or random behaviours. The main types of complexity are entropy, fractal dimension, and Lyapunov components [2]. We adopt in this work permutation entropy, which is defined as follows.

Let a time series (I_t) where each $I_t = (I_t^1, ..., I_t^l)$ and l is the number of components in the series. Consider the set P_n of permutations π of order n. For each block of patterns instances of size T, the relative frequency of permutations π in the block $I_t = (I_1^j, ..., I_T^j)$ was defined according to [2]:

$$p^j(\pi) = \frac{\sharp(t/0 \leq t \leq T-n, (x_{t+1}^j, ..., x_{t+n}^j) \; has \; type \; \pi)}{T-n+1}$$

where $j \in \{1, ..., l\}$.

We will say that $(I_{t+1}^j, ..., I_{t+n}^j)$ was type of π if it is ordered in a similar manner than π. For example, (I_{t+1}^j, I_{t+2}^j) was type of $\pi_{01}(01)$ (resp.$\pi_{10}(10)$) if $(I_{t+1}^j < I_{t+2}^j)$ (resp. $(I_{t+1}^j > I_{t+2}^j)$).

Definition 1 (Permutation Entropy). *The permutation entropy of order n of block patterns instances of size T is defined by :*

$$H(n) = -\sum_{j=1}^{l} \sum_{\pi \in P_n} p^j(\pi) \log(p^j(\pi))$$

The permutation entropy of order n, noted $H(n)$, obtained by definition 1 is the information contained in the block patterns of instances by comparing consecutive values of this block.

The measures of permutation entropy of consecutive and tangled blocks patterns instances of sized T in a time series of size s provides a signal of size $(s - T + 1)$, called KIR signal (for Killer Inhibition Reaction) and noted sig_{kir}. Such a signal is the information contained in the behaviour of instances of the studied time series.

KAR-operator

The third operator, the operator called KAR and noted op_{kar}, modeled in phase 3, the mechanism of activation of NK cells by Damerau-Levenshtein distance DDL between two consecutive patterns of instances [17]. This distance represents the affinity between two consecutive patterns of instances and calculates the required minimum number of insertions, deletions, substitutions and transpositions in the transformation of a permutation to another. Levenshtein distance between two patterns instances I_1 and I_2 says :

$$DDL(I_2, I_1) = DDL(I_2 \rightarrow I_1) = argmin(N_S \cdot P_S + N_O \cdot P_O + N_I \cdot P_I)$$

N_S, N_O and N_I are respectively the number of substitutions, omissions and insertions and P_S, P_O and P_I the positive or zero weight associated with each of these operations.

The choice of DDL distance is justified by the fact that it minimizes the mistake in comparisons between the patterns of instances because it uses a single edit operator transposition instead of two [5]. This operator calculates the affinity of the instances behaviours of the time series and allows processing patterns of instances for activation and as a result provides an activation signal called KAR (Killer Activation Reaction) and noted $sig_{kar} = (DDL((I_1, I_2), ..., DDL(I_{s-1}, I_s))$ where s is the length of the time series.

Balance-operator

The fourth operator, the operator called Balance and noted $op_{balance}$, models in phase 4, the competitive comparisons between inhibition and activation signals of the NK cell mechanism. This operator can distinguish between normal and abnormal behaviour in a studied time series by comparing the signals sig_{kir} and sig_{kar} with an estimate of their spectral density (PSD). For each signal, the estimation method used in NK system is MUSIC algorithm (MUltiple SIgnal Classification), which is based on the calculation of the inverse of the projection of eigenspaces of the matrix of auto correlation of the signal on noise subspace [22]. Therefore, the main feature of the MUSIC algorithm is to eliminate noise in the signal and to distinguish the clusters of the signal by projection on the noise subspace of the signal.

The basics of MUSIC algorithm

MUSIC algorithm is essentially a method of characterizing the range of a self-adjoin operator, in our case; we form the correlation matrix $A_{n,m} = E(x_n\overline{x_m})$ where E denotes the expected value and x is sig_{kir} or sig_{kar}. A is the self-adjoin operator of sig_{kir} or sig_{kar}. Suppose A is a self-adjoin operator with eigenvalues $\lambda_1 \geq \lambda_2$.... and corresponding eigenvectors υ_1, υ_2.... Suppose the eigenvalues $\lambda_{M+1}, \lambda_{M+2}$.... are all zero, so that the vectors $\upsilon_{M+1}, \upsilon_{M+2}$.... span the null space of A. Alternatively, $\lambda_{M+1}, \lambda_{M+2}$.... could merely be very small, below the noise level of the system represented by A; in this case we say that the vectors $\upsilon_{M+1}, \upsilon_{M+2}$.... span the noise subspace of A. We can form the projection onto the noise subspace; this projection is given explicitly by (eq1) : $P_{noise} = \sum_{j>M} \upsilon_j \overline{\upsilon_j}^T$ where the superscript T denotes transpose, the bar denotes complex conjugate, and $\overline{\upsilon_j}^T$ is the linear functional that maps a vector f to the inner product $\langle \upsilon_j, f \rangle$. The (essential) range of A, meanwhile, is spanned by the vectors $\upsilon_1, \upsilon_2, ..., \upsilon_M$ The key idea of MUSIC is this: because A is self-adjoin, we know that the noise subspace is orthogonal to the (essential) range. Therefore, a vector f is in the range if and only if its projection onto the noise subspace is zero, i.e., if $\|P_{noise} f\| = 0$ and this, in turn, happens only if (equ2) : $\frac{1}{\|P_{noise} f\|} = \infty$ equation 2 is the MUSIC characterization of the range of A. We note that for an operator

that is not self-adjoin, MUSIC can be used with the singular value decomposition instead of the eigenvalue decomposition.

By analogy with the biological mechanism of NK cells that detects attacks, based on the comparison of changes in signal activation and inhibition, we verified empirically that the differentiation of each PSD produced by MUSIC algorithm, which provides information variation on the energy contained in each signal, highlighted aberrant behaviour in a range of frequencies where the absolute derived of each PSD sig_{kir} and sig_{kar} signals are almost null.

4 Application

In this application, we are looking at the detection of fraud behaviour in mobile telephony. The types of fraud most studied are fraud intermediaries and subscription fraud. Fraud intermediaries are motivated by the bonuses paid by the Telecom operators to intermediate traders to acquire new subscribers, and retain its subscribers [13]. To take advantage of these bonuses, the intermediate traders must register new subscribers. The intermediate trader records fraudster fictitious subscribers, and makes short duration calls on the account of each of these subscribers. In addition, subscription fraud occurs when a subscriber uses a service with a false identity in order not to pay his bill. Both types of fraud usually can be identified by a large number of calls, often to international destinations. The existing algorithms for detecting the intrinsic behaviour of subscribers such as [4] and [16] require a large amount of data for learning and are sensitive to initial conditions.

We apply our NK system on simulated data telecommunications relating to the trafficking of some users from an intermediate trader whose proportion of the operations of fraud represents 0.01% of the sample simulated. These data include 10100 instances that represent the daily operations of an intermediate traders fraudster and express six attributes such as: the called and calling number, codes relays BTS, the duration of the call and the cost of call. To express an instance of behaviour of an intermediate trader in the day, these variables are aggregate for a day of the operation [13]. These aggregates are of two types: qualitative and quantitative. The binary representation of the time series is produced by transforming quantitative variables to percentiles modalities of order 4 and binarising each modality of qualitative variables generating 10100 instances and 52 components. The implementation of Phase 2 for the generation of a signal inhibition sig_{kir} requires adjustment parameter n corresponding to the order of permutations used in calculating the permutation entropy. In our application, this parameter varies between 2 and 5 providing four signals sig_{kir}.

Detecting fraud behaviour of the intermediate trader is based on the calculation of spectral densities of PSD sig_{kar} and $sig_{kir}$and then calculating the differentiation of this PSD denoted der_{kir} and der_{kar}. Firstly, the simulations permit to test the impact of the variation of the order of the permutations on the zone of detection of fraudulent behaviours. Secondly, we study the variation and the performance of NK system to detect the zone of fraudulent behaviour.

Indeed, for every choice of a set of orders of permutations $n_i (i = 2, 3, 4, 5)$, the NK system has been tested randomly on 100 different samples of affected data. The table 1 regroups the different descriptive statistics of der_{kir} :

Table 1. Descriptive statistics of der_{kir} and CPU is time of execution in second

	$min(der_{kir})$	$max(der_{kir})$	$std(der_{kir})$	CPU
n_2	0.0003	7.1056	0.9554	1.97
n_3	0.0003	7.1237	0.9539	1.98
n_4	0.0003	7.0892	0.95	2.01
n_5	0.0003	6.9767	0.9412	2.02

These different simulations showed that the variations of the orders of permutations don't influence on the zone of detection of fraudulent behaviour.

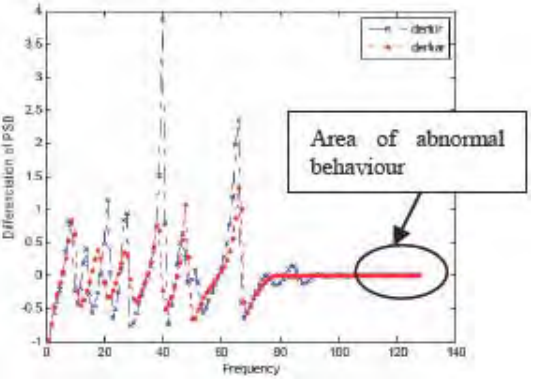


Fig. 2. Derivative of the PSD of the sig_{kir} and sig_{kar}

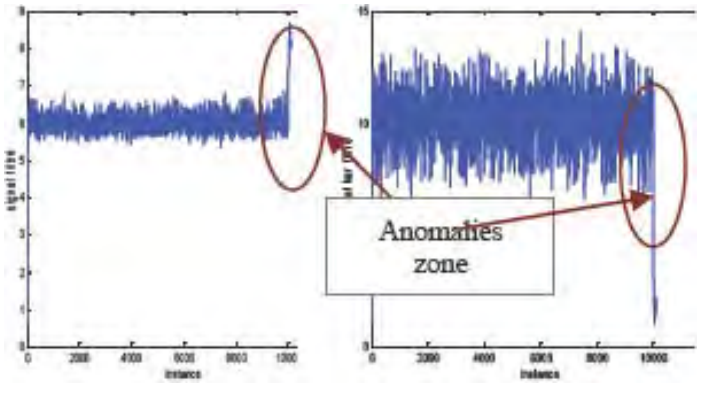


Fig. 3. To the left is sig_{kir} filtered and to the right is sig_{kar} filtered

Table 2. Comparative table between the different approaches for unsupervised Learning to fraud detection in mobile telephony

	SOM	Cluster hierarchic	K-Means	NK systme
False positive	0.019%	12.842%	22.727%	0%
False negative	0%	87.185%	77.273%	7%

The results are satisfactory because, despite the very low proportion of fraudulent transactions in the sample, NK system was able to detect (see figure 2 for an example of a sample), and to identify instances of fraud by a finite impulse response (FIR) filter (see figure 3).

These results are compared with other unsupervised algorithms such as self organisation map (SOM), K-Means and hierarchic clustering. The table 2 shows that NK system and SOM are performing to detect the abnormal behaviour. To study the reliability of the detection of each algorithm, the classes of each algorithm are compared with the variable indicator of fraud that is not included in the input variables for each algorithm.

5 Conclusion

We have proposed in this paper a new Artificial Immune System (AIS) called NK system for the detection of the existence of abnormal behaviour with an unsupervised approach. Its originality lies in the unsupervised detection based on the mechanism of NK cell (Natural Killer cell) unlike existing AIS using supervised approaches based on the mechanisms of T and B cells. The algorithm of our NK system has four main phases concerning the recognition and extraction models indictment (behaviours), transforming them into inhibition and activation signals, which will be placed in a final phase in order to detect the presence of abnormal behaviour.

We applied our NK system to detect the existence of fraudulent intermediary of a telecom operator. The empirical results show the effectiveness of our system to detect the presences of fraudulent transactions despite their low proportion in our sample. We will test in the future our system NK on a greater number of simulated data as well as real data from a Moroccan telecommunications operator.

References

1. Aickelin, U., Cayzer, S.: The danger theory and its application to artificial immune systems. In: The proceedings of 1st International Conference on Artificial Immune Systems (ICARIS), University of Kent at Canterbury, UK, pp. 141–148 (2002)
2. Bandt, C., Prompe, B.: Permutation entropy - a natural complexity measure for time series. Phys. Rev, Lett. 88, 174102 (2002)
3. Burnet, F.: The clonal selection theory of acquired immunity. Vanderbilt University Press, Nashville (1959)
4. Cahill, M., Lambert, D., Pinheiro, J.: Detecting fraud in the real world. Handbook of Massive Datasets (2002)

5. Damerau, F.J.: A technique computer detection and correction of spelling errors. Communications of ACM 7(3), 171–176 (1964)
6. Darmoul, S., Pierreval, H., Gabouj, S.: Scheduling using artificial immune system metaphors: A review. In: IEEE International Conference on Service Systems and Service Management, Troyes France, pp. 1150–1155 (2006)
7. Dasgupta, D., Forrest, S.: Artificial immune system in industrial application. In: International conference on Intelligent Processing and Manufacturing Material (IPMM), Honolulu, HI, vol. 1, pp. 257–267 (1999)
8. De Castro, L., Timmis, J.: Artificial immune systems: A new computational intelligence approach. Springer, Heidelberg (2002)
9. Farag, S.S., et al.: Nk receptors: Biology and clinical relevance. Blood 100(6), 1935–1947 (2002)
10. Fred, K.G., Marengo, E.A., Devanery, A.J.: Time-reversal imaging with multiple signal classification considering multiple scattering between the targets. J. Acoust. Soc. Am. 115(6), 3042–3047 (2004)
11. Garrett, S.M.: How do we evaluate artificial immune systems? Evolutionary Computation 13(2), 145–178 (2005)
12. Goldsby, R., Kindt, T., Osborne, B.: Kuby Immunology, 4th edn. WH Freeman at MacMillan Press (2000)
13. Heerden, J.H.V.: Detection fraud in cellular telephone networks. Master's thesis, Interdepartmental program of operational Analysis. University of Stellenbosch, South Africa (2005)
14. Hofmeyr, S.A.: An interpretative introduction to the immune system. In: Design Principles for the Immune System and Other Distributed Autonomous Systems ed., Oxford University Press, Oxford (2000)
15. Hofmeyr, S.A., Forrest, S.: Architecture for an artificial immune system. Journal of Evolutionary Computation 7(1), 45–68 (1999)
16. Hollmen, J., Tresp, V.: Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In: Kearns, M., Solla, S., Cohn, D. (eds.) P. of the Conference (NIPS 11). Advances in Neural Information Processing Systems 11, pp. 889–895. MIT Press, Cambridge (1998)
17. Hyyrö, H.: A bit-vector algorithm for computing levenshtein and damerau edit distances Nord. J. Comput. 10(1), 29–39 (2003)
18. Jerne, N.K.: Towards a Network Theory of the Immune System, vol. 125C, pp. 373–389. Annales d'Immunologie (1974)
19. Kantardzic, M.: Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons, Chichester (2003)
20. Lee, R.B., Shi, Z., Yang, X.: Efficient permutation instructions for fast software cryptography. IEEE Micro. 21(6), 56–69 (2001)
21. Luh, G.-C., Liu, W.W.: An immunological approach to mobile robot reactive navigation. Applied Soft Computing Journal (2006); 10.1016/j.asoc.2006.10.009
22. Madisetti, V.V., Douglass, B.: The digital signal processing handbook. IEEE Press, CRC Press (1988)
23. Matzinger, P.: The danger model in its historical context. Scandinavian Journal of Immunology 54, 4–9 (2001)
24. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied cryptography. CRC Press, Inc., Boca Raton (1996)
25. Twycross, J., Aickelin, U.: Towards a conceptual framework for innate immunity. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 112–125. Springer, Heidelberg (2005)

Markov Chain Analysis of Genetic Algorithms Applied to Fitness Functions Perturbed by Multiple Sources of Additive Noise

Takéhiko Nakama

Department of Applied Mathematics and Statistics
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218 USA
nakama@jhu.edu

Summary. We investigate the convergence properties of genetic algorithms (GAs) applied to fitness functions perturbed by multiple sources of additive noise that each take on finitely many values. Our analysis of GAs in noisy environments employs a novel approach; we explicitly construct a Markov chain that models such GAs and analyze it to uncover the convergence properties of the algorithms. We show that the chain has only one positive recurrent communication class. This immediately implies that the GAs eventually (i.e., as the number of iterations goes to infinity) find at least one globally optimal solution with probability 1. Furthermore, our analysis shows that the chain has a stationary distribution that is also its steady-state distribution. Using this property and the transition probabilities of the chain, we derive an upper bound for the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

Keywords: Genetic algorithms, Markov chains, noisy environments.

1 Introduction and Summary

Random noise perturbs objective functions in a variety of practical optimization problems, and genetic algorithms (GAs) have been widely proposed as an effective optimization tool for dealing with noisy objective functions (e.g., [2], [3]). Theoretical studies that examine evolutionary computation schemes applied to perturbed fitness functions typically assume that fitness functions are disturbed by a single source of additive noise (e.g., [1], [2], [5], [6], [12], [13]). In this study, we examine GAs applied to fitness functions perturbed by multiple sources of additive noise. In many optimization problems, objective functions may be perturbed by more than one noise source. For example, if objective function values are measured by a device that consists of multiple components, then each component may independently disturb objective function evaluations. Similarly, if objective function evaluations are conducted through several stages, then they may be perturbed differently at different stages. Clearly, disturbance by a single source of noise is in general stochastically quite different from that by multiple

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 123–136, 2008.
springerlink.com

sources of noise. We believe that our study is the first to rigorously examine the transition and convergence properties of GAs applied to fitness functions perturbed by multiple noise sources.

For analytical tractability, we assume that the multiple sources of additive noise each take on finitely many values and that they independently disturb fitness functions. However, we do not assume that they are identically distributed, and neither do we make any assumptions about their expected values or variances. We fully characterize this noisy environment in Sect. 3.

We take a novel approach to investigating GAs in noisy environments; we explicitly construct a Markov chain that models GAs applied to perturbed fitness functions and analyze it to characterize the transition and convergence properties of the algorithms. Although quite a few studies (e.g., [1], [2], [5], [6], [12], [13]) have examined GAs and more general evolutionary algorithms in noisy environments using either numerical or other theoretical methods, Markov chain analysis has not been applied to such GAs. (For a survey of the literature, we refer the reader to [2] and [8].) This is quite contrastive to the noiseless case; Markov chain theory has been effectively used to reveal important properties of GAs applied to noiseless fitness functions (e.g., [4], [14], [15], [16], [17], [18]). Markov chains considered to analyze the noiseless case typically have a state space that consists of distinct populations. However, these Markov chains fail to explicitly capture the evolution of GAs in noisy environments. The construction of a Markov chain for the noisy case is fully explained in Sect. 4. We explicitly compute the transition probabilities of the chain; see Theorem 1. It turns out that these probabilities can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

We show that the Markov chain for the noisy environment is indecomposable; it has only one positive recurrent communication class. This is Theorem 2. It follows immediately from this theorem that GAs eventually (i.e., as the number of iterations goes to infinity) find at least one globally optimal solution with probability 1. Theorem 3 states that the chain has a stationary distribution that is also its steady-state distribution. Based on this property and the transition probabilities of the chain, Theorem 4 provides an upper bound for the number of iterations sufficient to ensure with certain probability that a GA has reached the set of globally optimal solutions and continues to include in each subsequent population at least one globally optimal solution whose observed fitness value is greater than that of any suboptimal solution. We describe the details of these results in Sect. 5.

2 Preliminaries

We assume that the search space S consists of 2^L binary strings of length L as in traditional GAs. These 2^L candidate solutions are also referred to as chromosomes. Since the search space is finite, these chromosomes will be labeled by

integers $1, \ldots, 2^L$. Let f denote the (noiseless) fitness function, and let S^* denote the set of chromosomes that are globally optimal solutions:

$$S^* := \{i \in S \mid f(i) = \max_{j \in S} f(j)\} \ . \tag{1}$$

Then the objective of GAs is to find $i \in S^*$.

The implementation of GAs considered in this study consists of the following steps:

1. An initial population of M chromosomes is formed, and the fitness of each of the chromosomes is evaluated.
2. A chromosome with the highest observed fitness value is determined (the elitist strategy includes this chromosome in the next population).
3. Selection is performed on the chromosomes in the current population, and $\frac{M-1}{2}$ pairs of chromosomes are formed (M is assumed to be odd).
4. Crossover is performed on each of the pairs to generate offspring.
5. Mutation is performed on each offspring, and this completes the formation of the next population of size M.
6. The fitness of each of the new chromosomes is evaluated. If this population satisfies some stopping criterion, then the algorithm terminates. Otherwise, steps 2–6 are repeated.

GAs are assumed to implement the elitist strategy. This guarantees that the best candidate solution in the current population, which is a chromosome with the highest *observed* fitness value, is included in the next population. Hence it is important to realize that the elitist strategy guarantees the monotonic improvement of (noisy) *observed* fitness but may fail to monotonically improve (noiseless) fitness.

Another important strategy is that in step 6, GAs reevaluate the fitness value of each population member except for that of the chromosome preserved by the elitist strategy every time a new population is formed. It will become clear that this reevaluation of the fitness value is also essential for ensuring that GAs eventually find at least one globally optimal solution.

When GAs terminate, note that they select a chromosome that has the highest *observed* fitness value among the chromosomes in the last population as a (candidate for a) globally optimal solution. Since the observed fitness value may not be the same as the (noiseless) fitness value of the chromosome due to noise, GAs may not choose a globally optimal solution even if it is included in the last population; in order for GAs to correctly identify a globally optimal solution contained in the last population, the chromosome must have the highest observed fitness value. This observation is essential for properly characterizing GAs in noisy environments.

In Sect. 3, we specify the details of the genetic operations described in this section and mathematically define the noisy environment considered in this study. We construct a Markov chain that models these GAs and fully characterize its transitions in Sect. 4.

3 Mathematical Details of Noisy Fitness and Genetic Operations

Although we attempt to closely follow the notation developed by previous studies that conducted Markov chain analysis of GAs (for example, see [4], [14], [15], [16], [17], [18]), our notation inevitably becomes complicated due to the inclusion of noise. However, all of our new notation extends the conventional notation rather naturally.

First, we select M chromosomes from S *with replacement* to form an initial population $\mathcal{P}_0$. The population generated during the k-th iteration (the k-th population) will be denoted by $\mathcal{P}_k$. Let $m(i, \mathcal{P}_k)$ denote the number of instances of chromosome i included in the k-th population $\mathcal{P}_k$. Note that $\mathcal{P}_k$ is a multiset of chromosomes for each k. Let $i(j, \mathcal{P}_k)$ represent the j-th instance of chromosome i in $\mathcal{P}_k$ (thus $1 \leq j \leq m(i, \mathcal{P}_k)$). We need this notation because we must distinguish all the elements in the multiset $\mathcal{P}_k$ in order to precisely characterize the mathematical properties of additive noise sources considered in this study and to define the states of the Markov chain we construct in Sect. 4.

At each iteration, GAs evaluate the fitness value of each chromosome in the population. We suppose that several sources of random noise additively perturb each fitness function evaluation. Hence if we let $F(i(j, \mathcal{P}_k))$ denote the observed fitness value of chromosome $i(j, \mathcal{P}_k)$ (the j-th instance of chromosome i in $\mathcal{P}_k$), then it can be written as

$$F(i(j, \mathcal{P}_k)) = f(i) + \sum_{d=1}^{D} X^{(d)}_{i(j,\mathcal{P}_k)} \ , \tag{2}$$

where the D random variables $X^{(1)}_{i(j,\mathcal{P}_k)}, X^{(2)}_{i(j,\mathcal{P}_k)}, \ldots, X^{(D)}_{i(j,\mathcal{P}_k)}$ represent D sources of additive noise that perturb the fitness value of $i(j, \mathcal{P}_k)$. Note that for each i, j, and $\mathcal{P}_k$,

$$f(i(j, \mathcal{P}_k)) = f(i) \ ,$$

because this is the deterministic component of the observed fitness value of chromosome i.

For analytical simplicity, we assume that each of the D noise sources $X^{(1)}_{i(j,\mathcal{P}_k)}$, $X^{(2)}_{i(j,\mathcal{P}_k)}, \ldots, X^{(D)}_{i(j,\mathcal{P}_k)}$ is discrete and takes on finitely many values. Thus for each d and $i(j, \mathcal{P}_k)$, we define $X^{(d)}_{i(j,\mathcal{P}_k)}$ by

$$X^{(d)}_{i(j,\mathcal{P}_k)} = x^{(d)}_n \text{ with probability } p^{(d)}_n \ , \quad 1 \leq n \leq N_d \ , \tag{3}$$

where N_d represents the number of distinct possible values of the d-th noise source $X^{(d)}_{i(j,\mathcal{P}_k)}$ (thus $N_d < \infty$ for each d). Here we do not assume that these noise sources are identically distributed. Thus they can have different numbers of possible values. However, we assume that these D noise sources are independent. We further assume that for each d, $X^{(d)}_{i(j,\mathcal{P}_k)}$ are independent and identically

distributed for all i, j, and $\mathcal{P}_k$; the d-th noise source has the same distribution for each observed fitness value, and it is independent of the d-th noise source (and all the other noise sources) in any other instance of fitness function evaluation.

Let $F^*_{\mathcal{P}_k}$ denote the highest observed fitness value achieved by the k-th population $\mathcal{P}_k$, and let $i^*(j^*, \mathcal{P}_k)$ represent an instance of a chromosome in $\mathcal{P}_k$ whose observed fitness value equals $F^*_{\mathcal{P}_k}$ (it is the j^*-th instance of chromosome i^* in $\mathcal{P}_k$). Then we have

$$F(i^*(j^*, \mathcal{P}_k)) = F^*_{\mathcal{P}_k} \geq F(i(j, \mathcal{P}_k)) \quad \forall\, i(j, \mathcal{P}_k) \in \mathcal{P}_k\ . \tag{4}$$

The elitist strategy guarantees the inclusion of $i^*(j^*, \mathcal{P}_k)$ in the next population $\mathcal{P}_{k+1}$. (If $F^*_{\mathcal{P}_k}$ is achieved by more than one $i(j, \mathcal{P}_k) \in \mathcal{P}_k$, then break a tie by selecting one of them uniformly at random to determine $i^*(j^*, \mathcal{P}_k)$).

Using the notation defined above, we describe the mathematical details of selection. At each iteration, selection is performed to form pairs of chromosomes. In this study, we consider forming $\frac{M-1}{2}$ pairs for concreteness (thus M is assumed to be odd). First, the fitness value of each chromosome in the current population $\mathcal{P}_k$ is evaluated. Each of the observed fitness values has the form shown in (2). GAs subsequently select $M-1$ chromosomes from $\mathcal{P}_k$ *with replacement* to form these pairs. Our Markov chain analysis of GAs in the noisy environment is valid for any selection scheme—for example, proportional selection, ranking selection, and tournament selection (e.g., [11], [12], [17]).

Crossover and mutation in our noisy case do not differ from those in the noiseless case. For each of the $\frac{M-1}{2}$ pairs formed by selection, crossover is performed in order to generate two new chromosomes from the pair. Due to the elitist strategy, crossover does not operate on the chromosome $i^*(j^*, \mathcal{P}_k)$ in (4). Similarly, the elitist strategy does not allow mutation to alter $i^*(j^*, \mathcal{P}_k)$. For each of the other chromosomes in $\mathcal{P}_k$, mutation inverts each bit of an individual chromosome with some predetermined probability μ. We assume $0 < \mu < 1$.

A new population $\mathcal{P}_{k+1}$ emerges upon completing selection, crossover, and mutation. The algorithm computes the fitness value of each chromosome in $\mathcal{P}_{k+1}$, and these steps are repeated until a stopping criterion is satisfied.

4 Framework of Markov Chain Analysis

Markov chains constructed to model GAs in the noiseless case typically have a state space that consists of all possible distinct populations that can be formed from S (for example, see [4], [16], [17]). However, these Markov chains fail to explicitly capture the evolution of GAs in noisy environments. Instead, we construct a Markov chain, call it (Z_k), whose state space consists of multisets not of chromosomes but of the *ordered* $(D+1)$*-tuples* defined below.

For each iteration of a GA in the noisy environment described in Sect. 3, the corresponding state of the chain (Z_k) can be derived from the population $\mathcal{P}$ as follows. We form M ordered $(D+1)$-tuples from the M chromosomes in $\mathcal{P}$ by pairing each chromosome $i(j, \mathcal{P})$ with the values of the D noise sources $X^{(1)}_{i(j,\mathcal{P})}$,

$X^{(2)}_{i(j,\mathcal{P})}, \ldots, X^{(D)}_{i(j,\mathcal{P})}$ observed when the fitness value of $i(j,\mathcal{P})$ is evaluated [see (2)]. We denote the resulting ordered $(D+1)$-tuple by

$$\left[i, X^{(1)}_{i(j,\mathcal{P})}, X^{(2)}_{i(j,\mathcal{P})}, \ldots, X^{(D)}_{i(j,\mathcal{P})} \right] , \tag{5}$$

where j and $\mathcal{P}$ are suppressed in the first entry because they are unnecessary. These M ordered $(D+1)$-tuples compose a state of the Markov chain for the noisy case [thus each state of this chain is a multiset of the ordered $(D+1)$-tuples].

It is important to recognize that in order to explicitly model GAs in the noisy environment using any Markov chain, we need the last D entries of each ordered $(D+1)$-tuples in (5) (or quantities that are mathematically equivalent to these entries) because the selection process of GAs applied to the noisy environment is based on a function of not only the chromosome but also the D noise sources defined at (3). We analyze the Markov chain (Z_k) to elucidate the transition and convergence properties of GAs in the noisy environment. We denote by $\mathfrak{T}$ the state space of (Z_k). Let $m(i,\mathcal{T})$ denote the number of instances of chromosome i included in the ordered $(D+1)$-tuples of $\mathcal{T} \in \mathfrak{T}$ (thus $m(i,\mathcal{T})$ is analogous to $m(i,\mathcal{P}_k)$ defined at the beginning of Sect. 3). Similarly, we denote by $m(x^{(d)}_n, \mathcal{T})$ the number of instances of the value $x^{(d)}_n$ of the d-th noise source contained in the ordered $(D+1)$-tuples of $\mathcal{T} \in \mathfrak{T}$. Note that for each $\mathcal{T} \in \mathfrak{T}$ and d,

$$\sum_{i=1}^{2^L} m(i,\mathcal{T}) = \sum_{n=1}^{N_d} m(x^{(d)}_n, \mathcal{T}) = M \, . \tag{6}$$

We are now ready to precisely characterize the transitions of the Markov chain (Z_k). The following theorem shows the exact transition probabilities of (Z_k). It turns out that these probabilities can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

Theorem 1. *Let (Z_k) denote the Markov chain with state space $\mathfrak{T}$ that models GAs in the noisy environment. Let $\mathcal{T}$ and $\mathcal{T}'$ denote states in $\mathfrak{T}$, and let $i^*(\mathcal{T})$ denote a chromosome in an ordered $(D+1)$-tuple of $\mathcal{T} \in \mathfrak{T}$ that has the highest observed fitness value. If the observed fitness value of $i^*(\mathcal{T}')$ is greater than or equal to that of $i^*(\mathcal{T})$, then for each k,*

$$\begin{aligned} P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} = (M-1)! \prod_{i=1}^{2^L} \frac{1}{\tilde{m}(i,\mathcal{T}')!} \phi(i,\mathcal{T})^{\tilde{m}(i,\mathcal{T}')} \\ \times \prod_{d=1}^{D} (M-1)! \prod_{n=1}^{N_d} \frac{1}{\tilde{m}(x^{(d)}_n, \mathcal{T}')!} (p^{(d)}_n)^{\tilde{m}(x^{(d)}_n, \mathcal{T}')} \, , \end{aligned} \tag{7}$$

where

$$\phi(i,\mathcal{T}) = P\{\textit{chromosome } i \textit{ is generated from state } \mathcal{T}\}\,, \tag{8}$$

$$\tilde{m}(i,\mathcal{T}') = \begin{cases} m(i,\mathcal{T}') - 1 & \textit{if } i = i^*(\mathcal{T}) \\ m(i,\mathcal{T}') & \textit{otherwise,} \end{cases}$$

and

$$\tilde{m}(x_n^{(d)},\mathcal{T}') = \begin{cases} m(x_n^{(d)},\mathcal{T}') - 1 & \textit{if } i^*(\mathcal{T}) \textit{ is paired with} \\ & x_n^{(d)} \textit{ in } \mathcal{T} \textit{ (and in } \mathcal{T}') \\ m(x_n^{(d)},\mathcal{T}') & \textit{otherwise.} \end{cases}$$

On the other hand, if the observed fitness value of $i^*(\mathcal{T}')$ *is less than that of* $i^*(\mathcal{T})$*, then*

$$P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} = 0\,.$$

Proof (of Theorem 1). To simplify some of the expressions involved in this proof, we let $\mathcal{C}(\mathcal{T})$ denote the set of chromosomes contained in the ordered $(D+1)$-tuples of $\mathcal{T} \in \mathfrak{T}$. Thus $\mathcal{C}(\mathcal{T})$ represents the population component of state $\mathcal{T}$ in $\mathfrak{T}$. For instance, in Example 4.1, we have $\mathcal{C}(\mathcal{T}) = \mathcal{P}_k$. Note that $\mathcal{C}(\mathcal{T})$ is unique and easily identifiable from $\mathcal{T} \in \mathfrak{T}$ since $\mathcal{C}(\mathcal{T})$ can be obtained from $\mathcal{T}$ by simply ignoring the second entry of each ordered $(D+1)$-tuple (5) in $\mathcal{T}$.

Clearly, the elitist strategy guarantees that $P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} = 0$ if the observed fitness value of $i^*(\mathcal{T}')$ is less than that of $i^*(\mathcal{T})$. Suppose that the observed fitness value of $i^*(\mathcal{T}')$ is greater than or equal to that of $i^*(\mathcal{T})$. For each k, we have

$$\begin{aligned} P\{Z_{k+1} = \mathcal{T}' | Z_k = \mathcal{T}\} &= P\{Z_{k+1} = \mathcal{T}', \mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}') | Z_k = \mathcal{T}\} \\ &= P\{Z_{k+1} = \mathcal{T}' | \mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}'), Z_k = \mathcal{T}\} P\{\mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}') | Z_k = \mathcal{T}\} \\ &= P\{Z_{k+1} = \mathcal{T}' | \mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}')\} P\{\mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}') | Z_k = \mathcal{T}\}\,. \end{aligned} \tag{9}$$

Here

$$\begin{aligned} &P\{\mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}') | Z_k = \mathcal{T}\} \\ &= P\{\text{each chromosome } i \text{ is generated } \tilde{m}(i,\mathcal{T}') \text{ times from } \mathcal{T}\} \\ &= (M-1)! \prod_{i=1}^{2^L} \frac{1}{\tilde{m}(i,\mathcal{T}')!} \phi(i,\mathcal{T})^{\tilde{m}(i,\mathcal{T}')}\,, \end{aligned} \tag{10}$$

because the random variables $\tilde{m}(i,\mathcal{T}')$ are distributed multinomially with parameters $M-1$ and $\phi(i,\mathcal{T})$ $(i = 1,\ldots,2^L)$. Note that $\tilde{m}$ is used instead of m since the elitist strategy ensures the inclusion of $i^*(\mathcal{T})$ in the next population. Also,

$$\begin{aligned} &P\{Z_{k+1} = \mathcal{T}' | \mathcal{C}(Z_{k+1}) = \mathcal{C}(\mathcal{T}')\} \\ &= P\{\text{each noise value } x_n^{(d)} \text{ appears } \tilde{m}(x_n^{(d)},\mathcal{T}') \text{ times in } \mathcal{T}'\} \\ &= \prod_{d=1}^{D} (M-1)! \prod_{n=1}^{N_d} \frac{1}{\tilde{m}(x_n^{(d)},\mathcal{T}')!} (p_n^{(d)})^{\tilde{m}(x_n^{(d)},\mathcal{T}')}\,, \end{aligned} \tag{11}$$

because for each d, the random variables $\tilde{m}(x_n^{(d)}, \mathcal{T}')$ are distributed multinomially with parameters $M-1$ and $p_n^{(d)}$ $(n = 1, \ldots, N_d)$. Combining (9)–(11), we obtain (7). □

The probabilities $\phi(i, \mathcal{T})$ defined at (8) depend on the selection scheme employed by GAs. They can be computed exactly.

5 Convergence Analysis

Using the Markov chain (Z_k) constructed in Sect. 4, we analyze the convergence properties of GAs applied to fitness functions perturbed by the multiple sources of additive noise described in Sect. 3. For each d, we arrange the N_d labels $x_1^{(d)}, x_2^{(d)}, \ldots, x_N^{(d)}$ representing the N_d possible values of $X_{i(j,\mathcal{P}_k)}^{(d)}$ defined at (3) in descending order:

$$x_1^{(d)} > x_2^{(d)} > \cdots > x_{N_d}^{(d)} . \tag{12}$$

This is simply for notational convenience. As defined at (13), S^* denotes the set of chromosomes that are globally optimal solutions. We have

$$f(i) \geq f(j) \quad \forall\, i \in S^*,\ \forall\, j \in S . \tag{13}$$

The following theorem guarantees that the GAs applied to the noisy environment eventually find at least one globally optimal solution with probability 1.

Theorem 2. *The Markov chain (Z_k) is indecomposable: It has only one positive recurrent communication class, which consists of states in $\mathfrak{T}$ that each contain at least one ordered $(D+1)$-tuple $[i, x_1^{(1)}, x_1^{(2)}, \ldots, x_1^{(D)}]$ with $i \in S^*$.*

Proof (of Theorem 2). From (12)–(13),

$$\begin{aligned} f(i) + \sum_{d=1}^{D} x_1^{(d)} &\geq f(j) + \sum_{d=1}^{D} x_{n_d}^{(d)} \\ \forall\, i \in S^*\ ,\ \forall\, j \in S\ ,\ \forall\, x_{n_d}^{(d)}\ &(1 \leq n_d \leq N_d)\ ,\forall\, d\ . \end{aligned} \tag{14}$$

Thus it follows from Theorem 1 that, starting from any state, the chain moves with positive probability to any state that contains at least one of the following $(D+1)$-tuples:

$$[i, x_1^{(1)}, x_1^{(2)}, \ldots, x_1^{(D)}]\ ,\quad i \in S^*\ . \tag{15}$$

Thus all states in $\mathfrak{T}$ that contain at least one such ordered $(D+1)$-tuple belong to the same communication class, which we denote by $\mathfrak{A}$.

Next we show that each state in $\mathfrak{T} \setminus \mathfrak{A}$ is transient. Since

$$f(i) > f(j) \quad \forall\, i \in S^*,\ \forall\, j \in S \setminus S^*\ ,$$

we have

$$f(i) + \sum_{d=1}^{D} x_1^{(d)} > f(j) + \sum_{d=1}^{D} x_{n_d}^{(d)}$$
$$\forall\ i \in S^*,\ \forall\ j \in S \setminus S^*,\ \forall\ x_{n_d}^{(d)}\ (1 \le n_d \le N_d)\ ,\forall\ d\ . \tag{16}$$

Therefore, once the chain hits any state in $\mathfrak{A}$, it follows from Theorem 1 and (16) that with probability 1, the chain never moves to any state that does not contain any of the ordered $(D+1)$-tuples shown in (15). Hence the states in $\mathfrak{T} \setminus \mathfrak{A}$ are all transient. Since the state space $\mathfrak{T}$ is finite, the communication class $\mathfrak{A}$ is positive recurrent, and the chain does not have any other positive recurrent communication class. □

Thus the chain hits the positive recurrent communication class $\mathfrak{A}$ and stays there with probability 1. Since each state in $\mathfrak{A}$ contains at least one globally optimal solution paired with noise values that maximize its *observed* fitness value, Theorem 2 immediately implies the following essential property of GAs in the noisy environment: With probability 1, the algorithms reach the set of globally optimal solutions and continue to include in each subsequent population at least one globally optimal solution whose *observed* fitness value is greater than that of any suboptimal chromosome. It is important that the globally optimal solution has the highest *observed* fitness value because GAs will otherwise fail to select it as a (candidate for a) globally optimal solution even if it is included in the last population. Thus GAs eventually find at leat one optimal solution with probability 1.

The next theorem follows from Theorem 2 and ensures the convergence of the chain to stationarity.

Theorem 3. *The Markov chain (Z_k) has a unique stationary distribution that is also its steady-state distribution: There exists a unique distribution π on $\mathfrak{T}$ such that*

$$\pi K = \pi\ ,$$

where K is the $|\mathfrak{T}| \times |\mathfrak{T}|$ transition kernel of (Z_k), and for any states $\mathcal{T}$ and $\mathcal{T}'$ in $\mathfrak{T}$,

$$\pi(\mathcal{T}) = \lim_{k \to \infty} P\{Z_k = \mathcal{T} | Z_0 = \mathcal{T}'\}\ .$$

Proof (of Theorem 3). From Theorem 2, the chain has only one positive recurrent communication class $\mathfrak{A}$, and states in $\mathfrak{T} \setminus \mathfrak{A}$ are all transient. Each state $\mathcal{A}$ in $\mathfrak{A}$ is aperiodic because the chain currently in state $\mathcal{A}$ stays in $\mathcal{A}$ after one transition with positive probability. Hence $\mathfrak{A}$ is an aperiodic positive recurrent class, and it is an elementary fact that such a chain has a unique stationary distribution that is also its steady-state distribution (for example, see Chapter 3 in [9] and Chapter 10 in [10]). □

The stationary distribution π in Theorem 3 satisfies

$$\pi(\mathcal{T}) > 0 \quad \forall\, \mathcal{T} \in \mathfrak{A}\,, \tag{17}$$

and

$$\pi(\mathcal{T}) = 0 \quad \forall\, \mathcal{T} \in \mathfrak{T} \setminus \mathfrak{A}\,. \tag{18}$$

Thus the number of nonzero entries in π equals $|\mathfrak{A}|$. Let $\pi^{(k)}$ denote the distribution of the chain (Z_k) at time k. From Theorem 3 and (17)–(18), we have

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\mathfrak{A}} \pi^{(k)}(\mathcal{T}) = 1\,, \tag{19}$$

which again shows that with probability 1, GAs eventually reach the set of globally optimal solutions and continue to include in each subsequent population at least one globally optimal solution whose observed fitness value is greater than that of any suboptimal solution. Thus, in order to determine how many iterations are sufficient to guarantee with certain probability that a GA selects a globally optimal solution upon termination, we need to analyze the convergence rate of (19). Clearly, (19) is equivalent to

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\mathfrak{T}\setminus\mathfrak{A}} \pi^{(k)}(\mathcal{T}) = 0\,, \tag{20}$$

and the remaining part of this section focuses on the convergence analysis of (20).

To analyze the convergence in (20), we need to establish more notation. Each ordered $(D+1)$-tuple $[i, x_{n_1}^{(1)}, x_{n_2}^{(2)}, \ldots, x_{n_D}^{(D)}]$ can be associated with its observed fitness value, which we denote by $F([i, x_{n_1}^{(1)}, x_{n_2}^{(2)}, \ldots, x_{n_D}^{(D)}])$:

$$F([i, x_{n_1}^{(1)}, x_{n_2}^{(2)}, \ldots, x_{n_D}^{(D)}]) := f(i) + \sum_{d=1}^{D} x_{n_d}^{(d)}\,. \tag{21}$$

Let W denote the number of distinct possible values of (21), and let $F_1, F_2, \ldots, F_W$ denote the W distinct values. For notational convenience, they will be arranged in descending order: $F_1 > F_2 > \cdots > F_W$. We define

$$H_j := \{\mathcal{T} \in \mathfrak{T} | \max_{[i, x_{n_1}^{(1)}, \ldots, x_{n_D}^{(D)}] \in \mathcal{T}} F([i, x_{n_1}^{(1)}, \ldots, x_{n_D}^{(D)}]) = F_j\}\,.$$

Thus, the highest observed fitness value of each state in H_j equals F_j. Note that H_1 denotes the set of states in $\mathfrak{T}$ that contain at least one ordered $(D+1)$-tuple $[i, x_1^{(1)}, x_1^{(2)}, \ldots, x_1^{(D)}]$ with $i \in S^*$; hence $H_1 = \mathfrak{A}$, and we have $\mathfrak{T} \setminus \mathfrak{A} = \bigcup_{j=2}^{W} H_j$. For the remaining analysis, we rewrite (20) as

$$\lim_{k\to\infty} \sum_{\mathcal{T}\in\bigcup_{j=2}^{W} H_j} \pi^{(k)}(\mathcal{T}) = 0\,. \tag{22}$$

We assign $|\mathfrak{T}|$ labels $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{|\mathfrak{T}|}$ to the $|\mathfrak{T}|$ states in $\mathfrak{T}$ as follows. The first $|H_1|$ labels $(\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{|H_1|})$ represent states that belong to H_1 (it does not matter exactly how those $|H_1|$ states are represented by the $|H_1|$ labels). The next $|H_2|$ labels $(\mathcal{T}_{|H_1|+1}, \mathcal{T}_{|H_1|+2}, \ldots, \mathcal{T}_{|H_1|+|H_2|})$ represent states that belong to H_2. We continue this process until we label all the states in $\mathfrak{T}$. With these labels, the i-j-th entry of the $|\mathfrak{T}| \times |\mathfrak{T}|$ transition kernel $K$of the Markov chain (Z_k) naturally represents the one-step transition probability from state $\mathcal{T}_i$ to state $\mathcal{T}_j$, and K is a block lower triangular matrix whose j-th diagonal block $K(j)$ is a $|H_j| \times |H_j|$ matrix $(1 \leq j \leq W)$. The eigenvalues of this transition kernel K provide bounds for the convergence rate of (22). Let $\sigma(K)$ denote the spectrum of the kernel. Since K is a block lower triangluar matrix, the eigenvalues of K are the eigenvalues of the W diagonal blocks $K(1)$, $K(2)$,..., $K(W)$:

$$\sigma(K) = \bigcup_{j=1}^{W} \sigma(K(j)) \; . \tag{23}$$

Let $\lambda_{j,l}$ denote the eigenvalues of the j-th diagonal block $K(j)$:

$$\sigma(K(j)) = \{\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,|H_j|}\} \; . \tag{24}$$

As stated in Theorems 2 and 3, the Markov chain (Z_k) has only one positive recurrent class, which is also aperiodic. Hence the eigenvalue 1 of the transition kernel K has multiplicity 1, and it belongs to $\sigma(K(1))$. Moreover, there are no other eigenvalues of modulus 1:

$$|\lambda| < 1 \quad \forall \; \lambda \in \sigma(K) \; , \; \lambda \neq 1 \; . \tag{25}$$

This is shown in [10] (see, in particular, Chapter 10).

We have $\pi^{(k)} = \pi^{(0)} K^k$, where $\pi^{(0)}$ is the initial distribution of the chain. Thus, the convergence rate of (22) can be analyzed by examining the $(|\mathfrak{T}| - |H_1|) \times (|\mathfrak{T}| - |H_1|)$ submatrix of K that is obtained by eliminating the first $|H_1|$ rows and the first $|H_1|$ columns of K. This submatrix will be denoted by $\tilde{K}$. From (23)–(25), we know that each entry of $\tilde{K}^k$ goes to zero as k approaches infinity (the modulus of each eigenvalue of $\tilde{K}$ is strictly less than 1), and the convergence rate of (22) is determined by how fast each entry of $\tilde{K}^k$ goes to zero. We have the following theorem:

Theorem 4. *There exists a constant $C < \infty$ such that for each k,*

$$\sum_{\mathcal{T} \in \bigcup_{j=2}^{W} H_j} \pi^{(k)}(\mathcal{T}) \leq C \lambda^{*k} \; ,$$

where $\lambda^* = \max\{|\lambda| : \lambda \in \bigcup_{j=2}^{W} \sigma(K(j))\}$.

Proof (of Theorem 4). This theorem is based on basic facts in matrix theory. We leave some of the elementary details to the reader; for example, see an excellent treatment of the subject in [7] (in particular Chapters 3 and 5). Let $\tilde{K}^k(q, r)$

denote the q-r-th entry of the k-th power of $\tilde{K}$. Considering the Jordan canonical form of $\tilde{K}$, one can show that for each k, $\tilde{K}^k$ satisfies

$$\tilde{K}^k(q,r) = \sum_{j=2}^{W} \sum_{l=1}^{|H_j|} \beta(q,r,j,l,k) \lambda_{j,l}^k \ ,$$

where $\beta(q,r,j,l,k)$ denote constants that can be derived from the Jordan canonical form of $\tilde{K}$. The $\lambda_{j,l}$ in (26) are defined at (24). Furthermore, using properties of the powers of Jordan blocks, one can find constants $\nu(q,r,j,l)$ such that for all k,

$$\tilde{K}^k(q,r) \le \sum_{j=2}^{W} \sum_{l=1}^{|H_j|} \nu(q,r,j,l) |\lambda_{j,l}|^k \ . \tag{26}$$

For each $\mathcal{T}_s \in \bigcup_{j=2}^{W} H_j$ (thus $s > |H_1|$), we also have

$$\pi^{(k)}(\mathcal{T}_s) = \sum_{q=1}^{|\mathfrak{T}|-|H_1|} \pi^{(0)}(\mathcal{T}_{|H_1|+q}) \tilde{K}^k(q, s-|H_1|) \ . \tag{27}$$

From (26)–(27), we obtain

$$\sum_{\mathcal{T} \in \bigcup_{j=2}^{W} H_j} \pi^{(k)}(\mathcal{T}) = \sum_{s=|H_1|+1}^{|\mathfrak{T}|} \pi^{(k)}(\mathcal{T}_s)$$
$$\le \sum_{s=|H_1|+1}^{|\mathfrak{T}|} \sum_{q=1}^{|\mathfrak{T}|-|H_1|} \pi^{(0)}(\mathcal{T}_{|H_1|+q}) \sum_{j=2}^{W} \sum_{l=1}^{|H_j|} \nu(q, s-|H_1|, j, l) |\lambda_{j,l}|^k \le C \lambda^{*k} \ ,$$

where

$$\lambda^* = \max\{|\lambda| : \lambda \in \bigcup_{j=2}^{W} \sigma(K(j))\} \ ,$$

and

$$C = \sum_{s=|H_1|+1}^{|\mathfrak{T}|} \sum_{q=1}^{|\mathfrak{T}|-|H_1|} \pi^{(0)}(\mathcal{T}_{|H_1|+q}) \sum_{j=2}^{W} \sum_{l=1}^{|H_j|} \nu(q, s-|H_1|, j, l) \ . \qquad \square$$

Note that $\lambda^* < 1$. With regard to (19), it follows from Theorem 4 that there exists a constant $C < \infty$ such that for each k,

$$\sum_{\mathcal{T} \in \mathfrak{A}} \pi^{(k)}(\mathcal{T}) \ge 1 - C \lambda^{*k} \ .$$

Thus λ^* basically determines the number of iterations sufficient to ensure with certain probability that a GA has reached the set of globally optimal solutions and continues to include in each subsequent population at least one globally optimal solution whose observed fitness value is greater than that of any suboptimal solution.

6 Discussion

To our knowledge, this study is the first to rigorously examine transition and convergence properties of GAs applied to fitness functions perturbed by multiple sources of additive noise. Our novel Markov chain analysis successfully shows that GAs eventually (i.e., as the number of iterations goes to infinity) find at least one globally optimal solution with probability 1 provided fitness disturbance is caused by additive noise sources each of which takes on finitely many values.

Our theoretical results probably have significant implications in practice. As mentioned in Sect. 1, objective functions may be disturbed by multiple independent additive noise sources in many practical optimization problems, and it is reassuring that GAs are guaranteed to find at least one globally optimal solution in the noisy environment. The fitness disturbance examined in this study may be considered fairly general. For example, note that no assumptions were made about the mean of each noise source; our results hold even if their expected values are nonzero.

Mathematically, a Markov chain is completely determined by its transition kernel, and we thoroughly described the kernel of the chain that models GAs in the additively noisy environment. Theorem 4 shows that the convergence rate of the probability in (19) or (20) is roughly determined by the maximum modulus λ^* of eigenvalues in the spectrum of the $(|\mathfrak{T}| - |H_1|) \times (|\mathfrak{T}| - |H_1|)$ submatrix $\tilde{K}$ of the transition kernel K described in Sect. 5. Both λ^* and C^* can be explicitly computed from the Jordan canonical form of $\tilde{K}$. Therefore, the transition probabilities of the chain can be effectively used to bound the number of iterations sufficient to ensure with certain probability that a GA selects a globally optimal solution upon termination.

Note that GAs considered in our study are assumed to reassess the fitness value of each population member except for that of the chromosome preserved by the elitist strategy every time a new population is formed. If the algorithms do not do this, then, examining the proof of Theorem 2, it is easy to see that the probability that they eventually find at least one globally optimal solution is less than 1. The reassessment of the fitness value also significantly reduces memory requirements compared to the strategy of storing a fitness value for every chromosome that has been included in some population formed since the beginning of the execution.

We are currently extending our Markov chain analysis to other noisy environments. For example, we are investigating properties of GAs applied to fitness functions perturbed by more general discrete noise or by continuous noise. We believe that our Markov-chain-theoretic approach to analyzing GAs in noisy environments will further elucidate essential theoretical and practical properties of the algorithms.

Acknowledgement. This research has been supported by the Acheson J. Duncan Fund for the Advancement of Research in Statistics.

References

1. Arnold, D.V.: Noisy Optimization with Evolution Strategies. Kluwer Academic Publishers, Boston (2002)
2. Beyer, H.G.: Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. Computer Methods in Applied Mechanics and Engineering 186, 239–267 (2000)
3. Chen, A., Subprasom, K., Ji, Z.: A simulation-based multi-objective genetic algorithm (SMOGA) procedure for BOT network design problem. Optimization and Engineering 7, 225–247 (2006)
4. Davis, T.E., Principe, J.: A Markov chain framework for the simple genetic algorithm. Evolutionary Computation 1, 269–288 (1993)
5. Di Pietro, A., White, L., Barone, L.: Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In: Proceedings of the 2004 Congress on Evolutionary Computation, vol. 2, pp. 1254–1261 (2004)
6. Goldberg, D.E., Rudnick, M.W.: Genetic algorithms and the variance of fitness. Complex Systems 5, 265–278 (1991)
7. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
8. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. IEEE Transactions on Evolutionary Computation 3, 303–317 (2005)
9. Karlin, S., Taylor, H.M.: A First Course in Stochastic Processes. Academic Press, New York (1975)
10. Karlin, S., Taylor, H.M.: A Second Course in Stochastic Processes. Academic Press, New York (1981)
11. Leung, K.S., Duan, Q.H., Xu, Z.B., Wong, C.K.: A new model of simulated evolutionary computation—convergence analysis and specifications. IEEE Transactions on Evolutionary Computation 5, 3–16 (2001)
12. Miller, B.L., Goldberg, D.E.: Genetic algorithms, selection schemes, and the varying effects of noise. Evolutionary Computation 4, 113–131 (1996)
13. Nissen, V., Propach, J.: On the robustness of population-based versus point-based optimization in the presence of noise. IEEE Transactions on Evolutionary Computation 2, 107–119 (1998)
14. Nix, A., Vose, M.D.: Modeling genetic algorithm with Markov chains. Annals of Mathematics and Artificial Intelligence 5, 27–34 (1992)
15. Rudolph, G.: Convergence analysis of canonical genetic algorithms. IEEE Transactions on Neural Networks 5, 96–101 (1994)
16. Suzuki, J.: A Markov chain analysis on simple genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics 25, 655–659 (1995)
17. Vose, M.D.: The Simple Genetic Algorithm. MIT Press, Cambridge (1999)
18. Vose, M.D., Liepins, G.E.: Punctuated equilibria in genetic search. Complex Systems 5, 31–44 (1991)

A Security Supervision System for Hybrid Networks

Francoise Sailhan[1], Julien Bourgeois[1], and Valérie Issarny[2]

[1] LIFC, University of Franche-Comté, Centre de Développement Multimédia, 1 cours Leprince-Ringuet 25201 Montbéliard, France
sailhan@ieee.org, julien.bourgeois@univ-fcomte.fr
[2] Arles project, INRIA-Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153, Le Chesnay Cédex, France
valerie.issarny@inria.fr

Summary. The traditional way of protecting networks and applications with e.g., firewalls and encryption, is no longer sufficient to protect effectively emerging hybrid wired-cum-wireless networks including *ad hoc* networks. Intrusion detection mechanisms should be coupled with preventive measures so as to identify unauthorised abuses. To this end, we propose a novel Hybrid Distributed Security Operation Center (HDSOC) which collects logs that are generated by any application/service, layer of the protocol stack or resource (e.g., router), providing a global view of the supervised system based on which complex and distributed intrusions can be detected. Our HDSOC further (i) distributes its capabilities and (ii) provides extensive coordination capabilities for guarantying that both the networks and the HDSOC components do not constitute isolated entities largely unaware of each others.

Keywords: Distributed intrusion detection, host-based and network intrusion detection, security and protection management, event notification.

1 Introduction

Until recently, most systems were operating over wired networks and were intended to be used as part of specific applications or in localised settings (e.g., building, campus or corporation). Spurred by the emergence of Wifi technologies and the advance of generalised eCommerce and pervasive applications (e.g., rescue and military application), interest has moved towards the provision of a global solution that interconnects in a secure manner changing sets of clients, services and networks. This construction of Internet-scale applications introduces new challenges consisting in securing large-scale wired-cum-wired networks, including Wifi-enabled *ad hoc* networks, which are spanning geographically dispersed sites and distinct administrative domains. The traditional way of protecting these networks and applications with e.g., firewalls and encryption is no longer sufficient due to the following reasons. First, *ad hoc* networks introduce security holes due to their vulnerability to a variety of factors e.g., open medium, cooperative algorithms. In addition, the best protection is always vulnerable to attacks due to unknown security bugs and improper configuration. It is therefore clear that

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 137–149, 2008.
springerlink.com

preventive measures should be coupled with intrusion detection mechanisms so as to identify unauthorised use and abuse. The Distributed Network Intrusion Detection Systems (DNIDSs) that have been proposed in the literature [1] [2], are extremely diverse in the mechanisms they employ to gather, analyse data and identify intrusion. However, DNIDSs share in common the fact that they glean intrusion data by monitoring the traffic and intercepting the network communications. More specifically, they mostly operate on the IP and transport layer headers and packets as well as the packet content, providing in depth packet analysis. Consequently, while DNIDSs are in a very convenient position wherein it has a complete access to all traffic traversing the managed network, their perspicacities suffer from:

- the cost (in term of processing usage) associated with the in depth analysis of the intercepted traffic. Note that one class of attacks commonly launched against DNIDS, lies in letting this DNIDS in a lethal state by spamming it with a large number of spurious traffic.
- the absence of information owned by the DNIDS on resources (hosts, services, protocols and applications) that constitute the network, which renders the DNIDS impotent to detect, correlate and report a wide range of (host, service, protocol and application-specific) intrusions.

This inefficiency of actual DNIDSs engaged us to propose a novel approach to intrusion detection, which were based on a Distributed Security Operation Center (DSOC)[6, 7]. Rather than relying exclusively on a resource-consuming and prone to attack traffic monitoring system, our DSOC collects logs that are generated by any application, service, DNIDS, layer of the protocol stack or resource (e.g., router) composing the managed system. As a result, our DSOC owns a global view of the supervised system - the state of any component being reported - based on which it can detect complex intrusions that are possibly originated by any component of the protocol/application stack and any hardware resource.

Based on this preliminary work, we propose a novel Hybrid DSOC (HDSOC) which is dedicated to provide intrusion detection in a large-scale hybrid network built upon wired-cum-wireless networks, which are geographically-dispersed and include Wifi-based *ad hoc* networks. Our HDSOC takes into account the specificities of *ad hoc* networks:

- Wireless hosts may operate under severe constraints e.g., limited bandwidth. This requires defining an HDSOC that reduces the overhead caused by its usage.
- The dynamics of hybrid networks including *ad hoc* networks, diminish the resilience of the HDSOC and necessitates to increase the decoupling of the HDSOC components so as to enable this latter to react and reconfigure in a timely way to network dynamics.

All these factors circumvent the need for supporting a global low-overhead HDSOC that is adapted to the network topology and characteristics (e.g., its dynamics, organisation) as well as the medium of communication which may be

e.g., unreliable and subject to unexpected disconnection. In order to increase the resilience of the HDSOC, we propose to (i) distribute its capabilities and (ii) provide extensive coordination capabilities for guarantying that both the networks and the HDSOC components do not constitute isolated entities largely unaware of each others. The proposed security operation center collects logs for detection and correlation without consuming significant network bandwidth while addressing missing, conflicting, bogus, and overlapping data. Further support for dynamic reassign correlation, and intrusion detection management responsibilities is provided to nodes as the topology evolves. The remainder of this paper is organised as follows. We introduce the proposed Distributed Security Operation Center (§ 2) before detailing each of its constitutive component (§ 3). Then, we conclude this paper with a summary of our results along with directions for future works (§ 4).

2 Design Rational

HDSOC aims to detect intrusion in a hybrid network composed of a collection of *ad hoc* networks which either (i) operate in physically isolated geographic sites (e.g., disaster-affected areas) or (ii) extend the coverage of e.g., public hot-spots, corporate buildings or large-scale urban areas. Each of these geographically dispersed *ad hoc* networks is further connected to a (wired) wide area network thanks to some gateway nodes (see Figure 1 for an overview of the network and the HDSOC architecture).

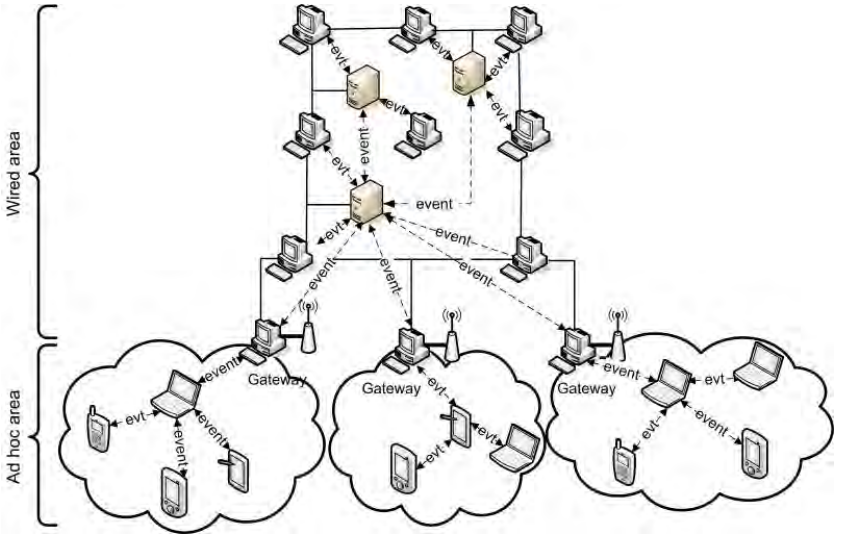


Fig. 1. HDSOC Architecture

The main challenge in collecting logs in *ad hoc* networks stems from the need to minimise the generated traffic and the computational load. This calls for:

1. parsing logs (i.e., extract only relevant data) rather than collecting raw logs that are characterised by a large size and prone to overload our system (as it is the case with attacks on the log size),
2. enabling resource-constrained devices that are incapable of parsing locally their logs, to delegate this parsing activity to a remote device which offers sufficient capabilities.
3. parsing logs as close as close as possible from the device that generates it so as to diminish the number of long distant communications.

In order to answer to these commitments, our approach lies in collecting and parsing logs locally whenever possible. For this purpose, a lightweight collector and parsing agent (hereafter referred as Embedded Collection Box or simply ECBox) is embedded on the devices that show sufficient memory and computing resources. Alternatively, for resource-constrained nodes, we rely on a service discovery protocol (§3.4), which discovers dynamically ECBoxes. Among potential candidates, the closest ECBox is selected so as to keep to a minimum long distance communication. The selected ECBox is further assigned the task of collecting and parsing logs on behalf of resource-constrained devices. After parsing and extracting relevant data from logs, local/remote ECBoxes generate event notifications that are further disseminated over the network, causing a slight increase of the network traffic due to the lightweight size of event notifications. The dissemination of events is performed by an event notification service which aims to ensure that each device is delivered the information (i.e., events) relating to the distributed intrusion to which that devices participates. This information gives to the device a global view of the intrusion (i.e., intrusion state, intrusion development and its level of implication), helping it in reporting in early stage any intrusion furtherance. In order to prevent the device from flooding the network whenever an intrusion is reported while providing to the devices a global view of the intrusion attempts to which it takes part, we rely on a publish/subscribe distributed event notification service whereby:

- consumers (e.g., devices taking part in the intrusion attempt, security administrator's computer) express their demands to producers (e.g., devices taking part in the intrusion) during a subscription process,
- event producers transfer to subscribers the description of any relevant event that has been triggered locally.

This event system faces the requirements (namely scalability, autonomy and timeliness) driven by HDSOC by disseminating in a distributed way events over a self-configured delivery structure organised as a cluster-based hierarchy (Figure 3). This cluster-based hierarchy provides convenient aggregation and correlation points while rending our HDSOC more adaptable to network failures and less vulnerable to attacks due to its distributed nature.

3 Distributed Operation Center for Hybrid Networks

We propose an hybrid distributed security operation center which attempts to detect intrusions within a large-scale network including geographically-dispersed wired or wireless networks (e.g., *ad hoc* networks). Such detection necessitates to collect logs (§3.1) so as to identify intrusion attempts (§3.2) and initiate proper reactions. Rather than collecting bandwidth-consuming raw logs, our approach lies in parsing logs so as to extract only security-relevant information and generate compact event notifications and alarms that are beside disseminated at low cost (§ 3.3). Note that such parsing is either performed locally, i.e., on the device that generated the logs if its capacities are sufficient) or alternatively remotely; such delegation of the parsing task being enabled by a pervasive service discovery protocol (§ 3.4).

3.1 Local Event Collection

Prior developing mechanisms for detecting intruders, it is crucial to understand the nature of attacks[1] as well as the possible security holes that characterise *ad hoc* and hybrid networks. In *ad hoc* networks, intruders take advantage of the lack of physical protection inherent to the absence of clear physical boundary (no protection being applied inside the network by any layer 3 resources, e.g., gateways), the collaborative nature of algorithms and the resource-limited capacity of the network (devices being more likely to be exhausted). This renders network components particularly vulnerable. These components include:

- *Networking components* (e.g., MAC, zeroconf and routing protocols) included in the physical/link/network layers are subject to eavesdropping, jamming, interceptions (physical layer), identity falsification (zeroconf protocol) and finally attacks initiated on routing tables (routing protocol) with e.g., the so-called wormhole and blackhole attacks.
- Security components, including DNIDS, cryptographic facilities, motivation functionalities which recompense collaborative nodes, and reputation and exclusion mechanisms whereby nodes vote and attribute a reputation to each node.
- Transport components, applications and services which are affected by session hijacking or flooding and application/service/scenario-specific attacks.

Each of these software components generates logs expressed in different formats including standard formats (e.g., syslog, MIB, HTML) or proprietary/application-specific formats. These logs can be easily collected relying on standard Xmit protocols (e.g., SNMP, SMTP, HTTP) as it is the case in traditional monitoring architectures. For instance, OLSR [5, 4] logs can be collected using MIB format that is defined and used in [8, 9]. The pervasiveness of these protocols ensures a significant level of interoperability to our HDSOC despite the heterogeneity of hardware

[1] Interest reader can refer to [12] for a comprehensive survey on attacks and countermeasures in *ad hoc* networks.

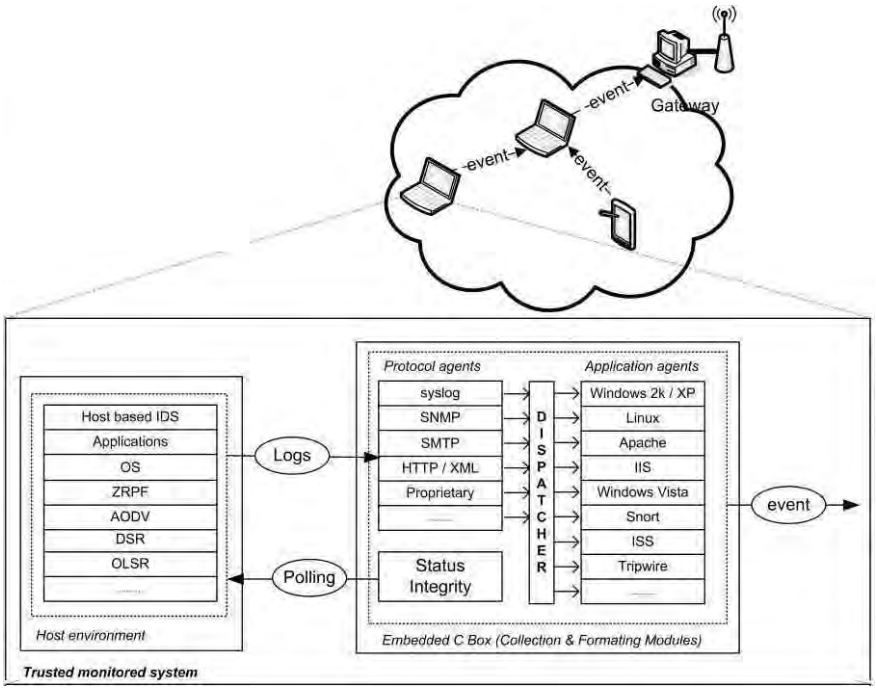


Fig. 2. Device Architecture

and software platforms. We therefore consider a HDSOC in which each device generates logs whose collection is enabled thanks to a protocol agent. In practice, this protocol agent corresponds to a collection of clients which implement standard Xmit protocols (Figure 2). Collecting logs from heterogeneous sources implies setting up a dispatcher and an application agent. The *dispatcher* determines the source-type of an incoming event and then forward it to the appropriate application agent which formats it in a common format (i.e., a format understandable by any HDSOC module). In practice, this dispatcher performs the following tasks:

- listening for incoming message transmitted by a protocol agent through an particular channel (e.g., as socket, named pipe, system V message queue),
- identifying the message source and the Xmit protocol used. More precisely, a patterns database is pre-loaded in the device memory (for performance considerations) and is used to find patterns matching the message.
- redirecting the original message to the application agent responsible for managing the messages generated by that type of source and Xmit protocol.

This application agent parses the message and expresses it in a common format that is transmitted to the event notification agent.

3.2 Distributed Intrusion Detection

After translating events into a common format that is understandable by any DSOC component, events are analysed and correlated so as to avoid transmitting all the events across the network. The main objective of correlation lies in producing a succinct overview of security-related activities. This necessitates to (i) filter events for the purpose of extracting only relevant events, and (ii) aggregate events so as to generate a compact representation of those events that eases the intrusion detection. Broadly sketched, event filtering aims to eliminate events that are not relevant, i.e.,

- duplicate events that do not provide additional information while consuming significant bandwidth.
- events that match policy criteria e.g., administrator login, identification, authorisation or restriction processes.
- events that are not critical to the supervised system, excluding events that relate to some vulnerabilities whose system is not exposed to. For this purpose, the device stores and maintains (structural, functional and topology-related) information about security breaches and insecure behaviour that either impact the overall security level or that can be exploited by an attacker.

Relevant events (i.e., events that went thought the filtering pipe) are further aggregated so as to provide a more concise view of what is happening in the system. This actual system view called *context* is stored locally with the previously generated contexts, before being transmitted by the event notification service. Based on the collection of contexts owned by the device, intrusion detection may take place. Intrusion detection consists in analysing a sequence of events so as to identify event sequence patterns characterising intrusion attempt. Note that during this process, time considerations are taken into account so as to take into account slow intrusions. In practice, such intrusion detection consists in matching a sequence of events (a context) against a set of attack signatures whose structure is described bellow.

Attack signature

A conquering attack can be broken down into a collection of successive steps that are successfully completed. This renders an attack characterisable as an attack signature, which corresponds to a labelled tree rooted by a node representing the goal, and intermediate nodes representing an attack step (i.e., an observable event) with a succession of children defining a way of achieving it. An attack scenario (i.e., the overall set of attacks that can threat the supervised system) is then represented as a forest of trees, with some part of trees being shared when a subset of steps involved in two distinct attacks is similar. Such attack scenarios are defined by the security administrator based on vulnerabilities specific to the network and the past attacks. Such tree-based representation of attack signatures renders intrusion detection easy to carry out. Indeed, an attack attempt is easily identifiable by matching attack signatures against a context, i.e., against a

succession of (possibly distributed) events occurring on a specific set of systems (e.g. devices, collection of devices, network segments). From a practical point of view, an attack identification therefore consists in matching an attack signature on the instance of a particular context. Central to intrusion identification is therefore the context accuracy. This accuracy is maintained by the event notification service, which updates the context of each device (its system view) with the most up-to-date events arising in the network.

3.3 Distributed Event Notification

Our event notification service aims to deliver events to devices so as to enable them to update their context, giving that device a global view of the intrusion attempts and hence rending the detection of intrusions more accurate. In order to prevent the device from blindly flooding the network whenever an intrusion is reported, our event model derives from the asynchronous publish/subscribe paradigm. From a communication perspective, our distributed event notification consists in exchanging notifications and control messages (i.e., subscriptions and un-subscriptions) between producers and subscribers through a collection of intermediate event agents (Figure 1). Note that a potential *event agent* (hereafter simply called agent) designates a device which holds our notification service. In practice, this collection of intermediate agents is organised into a cluster-based structure wherein each agent corresponds is a cluster leader and maintains information and connectivity with its cluster members and its clusterhead[2]. This underlying structure is then used for delivering control messages (subscriptions and un-subscriptions) to producers, as well as notifications to consumers. When delivering a notification, the main objective pursued by agents lies in forwarding that notification to an agent only if, toward this direction, there exists a consumer interested in receiving it. For the purpose of forwarding selectively notifications, each agent holds a subscription repository that includes each received subscription along with the respective neighbouring agent which forwarded it. Note that a neighbouring router constitutes the potential candidate for forwarding notifications. This repository is used to define if there exists a consumer along the direction of the considered router that subscribed for this notification. Based on this event notification, security information can be efficiently disseminated to the HDSOC.

Event notification Delivery Structure

In order to support an efficient event dissemination over a hybrid network, we propose a distributed event system, which distinguishes itself by:

- providing seamless integrated event notification over a network composed of different types of networking technologies (wired *versus* wireless, infrastructure-less *versus* infrastructure-based networks) and possibly spanning geographically dispersed domains/sites.

[2] The root of the delivery structure maintains information restricted to its cluster member.

- addressing the problem relating to the support of extensive control, security and autonomy by mean of a distributed event notification system based on a overlay infrastructure, which organise the nodes for delivering event notifications.

In practice, in order to distribute the events management, we based our event notification system on a distributed grouping communication which organises nodes into a self-organised delivery structure (Figure 3) deployed of the hybrid network, as defined in [3]. This structure corresponds to a cluster-based hierarchy of n_L layers (n_L=$log(n_n)$, with n_n designating the number of nodes that are expected to join the event system), each layer being portioned into a set of bounded-size clusters (let k be that size) controlled by a cluster head. The reason for setting bounds on the number of layers and on the cluster size is twofold. First, it ensures a control overhead ranging about $log(n_n)$ at each node. Second, the length of the path used for delivering notifications, and hence the related delay, is bounded by $o\ log(n_n)$. In order to prevent a cluster of size k from changing continuously its configuration whenever it gains or looses a member, the admitted bounded-size of a cluster ranges from k up to $2k$ cluster members. In practice, to warrant a loop-free structure, each node belongs to the lowest layer (L_0) and only the leader of cluster located in a layer L_i belongs to the upper layer L_{i+1}. This delivery structure is created and maintained by a grouping solution. Considering the fact that there exists no unique grouping protocol that is optimal for any kind of network, we use a specialised grouping solutions for each type of network along with a mechanism for integrating them. We

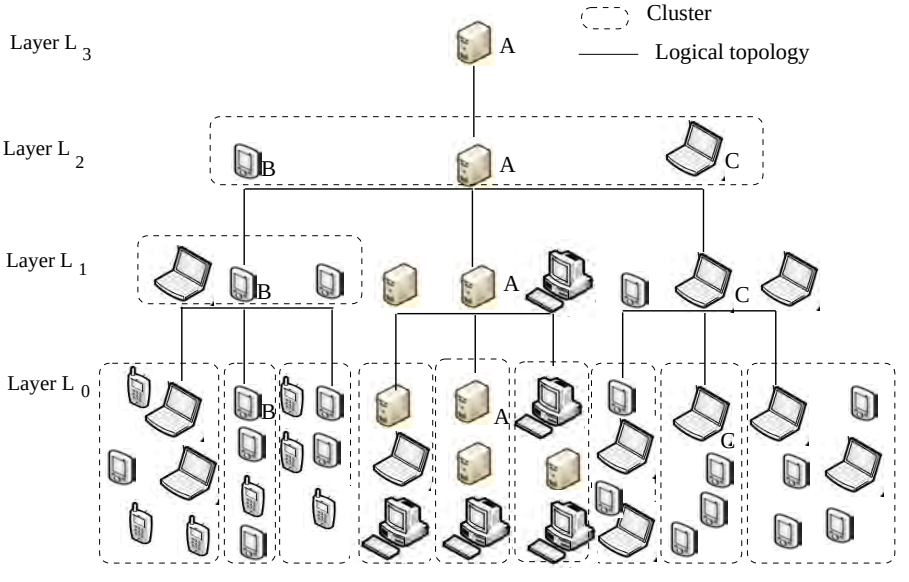


Fig. 3. Event Delivery Structure

rely on the Nice protocol [3] which has been specifically designed for operating in infrastructure-based large-scale network (e.g., the Internet) and the Madeira protocol that comes from our previous research on network management [10] and is customised to operating over *ad hoc* networks. The reason that motivates our choice for the Nice protocol, is twofold. First, this application-level protocol can operate over a large-scale network spanning different administrative domains. In addition, it were originally developed to support video streaming and hence meets the requirements driven by real-time delivery.

For scalability reasons, a node does not manage information concerning the overall group. Instead, a member or cluster head maintains information restricted to its cluster(s); each member sending periodically a *keep-alive* message. This limited knowledge permits to keep to a minimum the number of control messages exchanged for maintaining up-to-date membership information.

3.4 Distributed Service Discovery

The resource constraints of networked devices (e.g., routers, or devices belonging to *ad hoc* networks) coupled with the financial cost inherent to the deployment of additional functionalities on devices, circumvents the need for enabling HDSOC to delegate to remote devices a part of the functionalities relating to intrusion detection, e.g., log parsing and signature matching. The effective delegation of these functionalities, which are traditionally performed by ECBoxes, necessitates to discover on the fly the service(s) offered in the network that best match(es) these functionalities requirements. The following introduces a service discovery protocol [11] that meets this requirement. This protocol is aimed at hybrid networks including *ad hoc* networks. In the *ad hoc* network, our primary goal is to keep to a minimum the traffic generated by the service discovery process, so as to minimise consumption of resources and in particular energy. Specifically, our discovery architecture is structured around a subset of HDSOC nodes, called lookup agents or simply agents, that are responsible for discovering ECBox functionalities and capabilities (see Figure 4). These lookup agents are deployed so that at least one lookup agent is reachable in at most a fixed number of hops, $\mathcal{H}$, whose value is dependent upon the nodes density. Agents cache the descriptions of ECBoxes' functionalities (services) available in their vicinity which is defined by $\mathcal{H}$. Hence, HDSOC nodes (excluding lookup agents) do not have to maintain a cache of service descriptions, and the network is not flooded by service advertisements. A resource-constrained HDSOC device looking for a service (i.e., an ECBox or one of its embedded functionality), simply sends a query to the lookup agent for local service discovery. If the description of the requested service is not cached by the local agent, this agent selectively forwards the query to other lookup agents so as to perform global discovery. The selection of the lookup agents toward which service queries are forwarded, is based on the exchange of profiles among agents. The agent profile provides a compact summary of the agent's content and a characterisation of the host capacity. Agent profiles

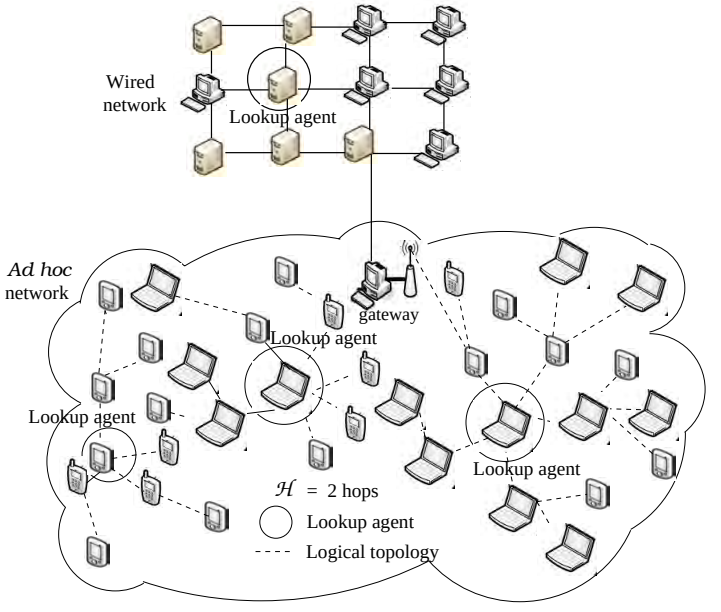


Fig. 4. Discovery Structure

allow both guaranteeing that service queries are issued to agents that are likely to cache the description of the requested service and to keep to a minimum the generated traffic. Another critical issue lies in providing convenient features so as to enable discovery of services over the hybrid network. This is supported through gateways that (i) advertise their capabilities (e.g., with a related gateway protocol, a zeroconf protocol, or using our protocol), and (ii) are assigned a local or remote lookup agent, implementing the cooperative behaviour as discussed above. A gateway lookup agent then holds the description of services available in all the networks composing the hybrid network, and advertises itself to the networks it bridges composing the hybrid network. Furthermore, to support service discovery in an infrastructure-based network in which a network administrator deploys agents, clients and service providers behave differently. Since clients and service providers do not need to elect a agents, they can listen for agents announcements or rely on the DHCP protocol.

Using this service discovery protocol, a resource-constrained DHSOC device can delegate to a remote device the resource-consuming functionalities that implement intrusion detection. For this purpose, it discovers dynamically the ECBoxes, located within the overall hybrid network, which offer the functionalities that best match its requirements. Then, it cooperates with the selected ECBoxe's service, utilising the provided service description, so as to contribute to the global effort for detecting intrusions.

4 Conclusion

In this paper, we propose a Hybrid Distributed Security Operation Center (HDSOC) which collects logs that are generated by any application/service, layer of the protocol stack or resource (e.g., router), providing a global view of the supervised system based on which it complex and distributed intrusions can be detected. Rather than directly transmitting these logs over the network, causing its overload, logs are parsed in an early stage so as to easily extract intrusion-related information and distribute it by the mean of compact event notifications and alarms. This HDSOC couples a lightweight distributed intrusion detection components with a distributed event system and a distributed service discovery protocol for an efficient delegation of resource-consuming tasks and a bandwidth-saving cluster-based collection of the events across the hybrid network. Our Hybrid Distributed Security Operation Center further addresses the main commitments of hybrid networks: scalability, flexibility, autonomy, and fault-tolerance. More precisely,

- *Scalability* comes from the distribution of load relating to the log parsing and the intrusion detection, on the devices
- *Autonomy* is the consequence of using a group based event notification service and a discovery protocol that are (i) automatically deployed without requiring human intervention and (ii) adapt dynamically to network changes (e.g., topology changes).
- *Fault-tolerance* is attributed to (i) a loosely-distributed event delivery that adapts dynamically to any permanent or transient network failure and a service discovery protocol that permits to discover dynamically an alternative to a faulty service.

Acknowledgements

Authors would like to acknowledge the implementation work carried by R. Bidou (University of Franche Comté), R. Chibout (INRIA), E. Cuadrado-Salamanca (EIRC, LM Ericsson Ltd), P. Farrell (EIRC, LM Ericsson Ltd) and A.K. Ganame (University of Franche Comté).

References

1. Anantvalee, T., Wu, J.: A survey on intrusion detection in mobile ad hoc networks. In: Wireless/Mobile Network Security. Springer, Heidelberg (2008)
2. Axelsson, S.: Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers University (2000)
3. Banerjee, S., Bhattacharje, B., Kommareddy, C.: Scalable application layer multicast. In: ACM SIGCOMM (2002)
4. CLausen, T., Jacquet, P.: Optimized link state routing protocol (olsr), rfc 3626 (October 2003), `http://www.ietf.org`

5. Clausen, T., Jacquet, P., Laouti, A., Muhlethaler, P., Quayyum, A., Viennot, L.: Optimized link state routing protocol. In: IEEE INMIC (2001)
6. Ganame, A.K., Bidoud, R., Bourgeois, J., Pies, F.: A high performance system for intrusion detection and reaction management. Journal of Information Assurance and Security (2006)
7. Ganame, A.K., Bourgeois, J., Bidou, R., et al.: A global security architecture for intrusion detection on computer networks. In: IEEE IPDPS (2007)
8. Pacheco, V., Puttini, R.: An administration structure for the olsr protocol. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part III. LNCS, vol. 4707. Springer, Heidelberg (2007)
9. Albers, P., Camp, O., Percher, J.M., et al.: Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. In: Wireless Information Systems (2002)
10. Sailhan, F., Fallon, L., Quinn, K., et al.: Wireless mesh network monitoring: Design, implementation and experiments. In: IEEE GLOBECOM-DANMS (2007)
11. Sailhan, F., Issarny, V.: Scalable service discovery for manets. In: IEEE PERCOM (2005)
12. Wu, B., Chen, J., Wu, J., Cardei, M.: A survey on attacks and countermeasures in mobile ad hoc networks. In: Wireless/Mobile Network Security. Springer, Heidelberg (2008)

Cost Minimization in Service Systems Using Particle Swarm Optimization

Tad Gonsalves and Kiyoshi Itoh

Department of Information and Communication Sciences,
Faculty of Science and Technology,
Sophia University, 7-1 Kioicho, Chiyoda-ku, Tokyo, 102-8554 Japan
{t-gonsal,itohkiyo}@sophia.ac.jp

Summary. This paper deals with the optimization of the operational costs of service systems. The cost function consists of service costs and waiting costs. Service cost is associated with the employment of service-providing personnel, while the waiting cost is associated with the customers having to wait for the service. The cost function is minimized subject to the server utilization as well as to the customer satisfaction constraints, using the Particle Swarm Optimization (PSO) algorithm. PSO is a fairly recent swarm intelligence meta-heuristic algorithm known for its simplicity in programming and its rapid convergence. The optimization procedure is illustrated with the example of a practical service system. A series of experiments show optimum results for the operation of the service systems.

Keywords: Service systems, Particle Swarm Optimization, PSO, Optimization.

1 Introduction

The goal of a service system is to provide service. In a service system, the providers of service are called servers and the recipients of service are called customers. Also known as value co-creation system, a service system is a configuration of technology and organizational networks. Post offices, banks, hospitals, reservation offices, schools and universities are some examples of practical service systems. The managerial authorities are often pressed to drastically reduce the operational costs of active and fully functioning service systems, while the system designers are forced to design (new) service systems operating at minimal costs. Both these situations involve system optimization.

Any optimization problem involves the objective to be optimized and a set of constraints [12]. In this study, we seek to minimize the total cost (tangible and intangible) to the system. The total cost can be divided into two broad categories - cost associated with the incoming customers having to wait for the service (waiting cost) and that associated with the personnel (servers) engaged in providing service (service cost). Waiting cost is the estimate of the loss to business as some customers might not be willing to wait for the service and may decide to go to the competing organizations, while serving cost is mainly due to the salaries paid to employees. If the organization decides to increase the level of

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 151–161, 2008.
springerlink.com

service provided, cost of providing services would increase, if it decides to limit the same, cost associated with waiting for the services would increase. So, the manager has to balance the two costs and make a decision about the provision of optimum level of service.

For each context in the system, the number of personnel to be assigned and the service time per customer are the decision variables. Although there are bounds on the decision variables, the number of combinations even for a modest service system is prohibitively large. The problem, therefore, is a typical combinatorial optimization problem that cannot be solved by an exhaustive search. We use the Particle Swarm Optimization (PSO) meta-heuristic algorithm to determine the values of the decision variables that optimize the cost function.

PSO is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [8-9]. The algorithm is inspired by the social behavior of bird flocking or fish schooling. It has been successfully applied to solving optimization problems in diverse disciplines. Compared to other evolutionary computational algorithms, PSO has many desirable characteristics. PSO is easy to implement, can achieve high-quality solutions quickly, and has the flexibility in balancing global and local exploration. More important, the memory mechanism of PSO can keep track of previous best solutions and, therefore, avoid the possible loss of previously learned knowledge [14].

We illustrate the simulation optimization technique using PSO, by means of a practical service system example shown in Fig. 1. The small clinic service system is made up of seven different 'contexts' represented by rectangles in the diagram. Reception, Diagnosis 1, Diagnosis 2, Prescription, Medical tests, Prescription and Accounts are the respective contexts. These contexts are essentially the places of work, or the service stations at which service is provided to the customers. The service-providing personnel include doctors, nurses, medical technicians, physiotherapists, pharmacists, etc. These, together with the rest of the resources are labeled above and below the contexts.

The service system model depicted in Fig. 1 has two aspects - static and dynamic. The static aspect shows the layout of the contexts and the 'hard-wired' connections among. There are also junctions such as branch (Br), join or serialize (Se), etc., which control the workflow in the system. These are represented by smaller rectangles. The modeling details of the service systems are found in [5]. The dynamic simulation gives life to the system. Bottlenecks and deadlocks in the system operation can be verified by means of the visual simulation. The service system model is functionally a queuing model, the operation of which is simulated by means of discrete event simulation [2, 4]. The simulation statistics include average queue lengths, server utilization, etc. which are used to evaluate the cost function and the constraints.

This paper is organized as follows: Section 2 presents the optimization problem formulation. Section 3 deals with the application of PSO to service systems optimization. Section 4 gives a brief conclusion and indications for further research.

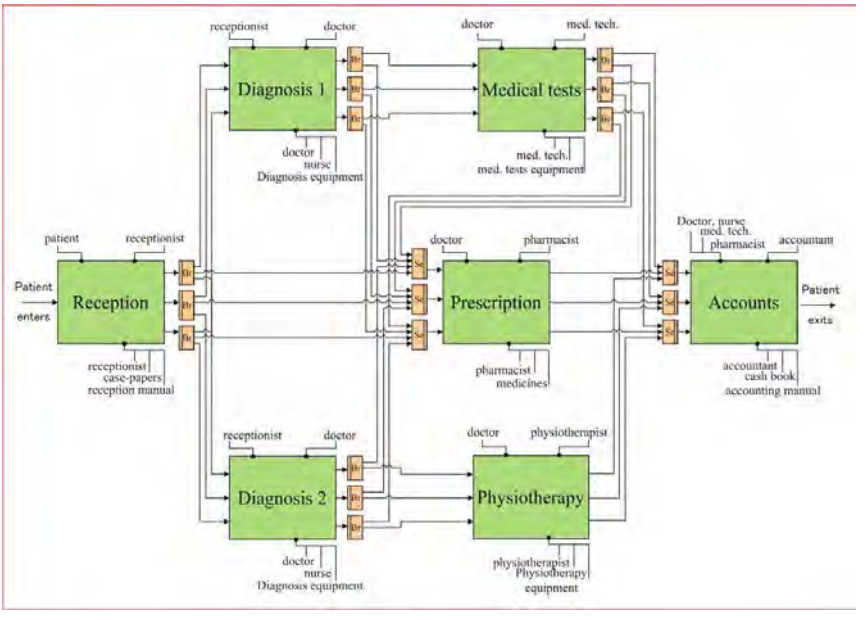


Fig. 1. Workflow model of a small clinic service system

2 Optimization Problem Formulation

2.1 Cost Function

To evaluate their performance, service systems are modeled and simulated as queuing networks. In a waiting line model, the total cost is estimated to be the sum of the waiting cost and the service cost [1, 6, 11] as shown in Fig. 2. Waiting cost is the estimate of the loss to business as some customers might not be willing to wait for the service and may decide to go to the competing organizations.

If Q_L is the average of the number of customers in queue and W_C is the waiting cost per customer per unit time, then the waiting cost per unit time is:

$$Waiting\,cost = Q_L W_C \tag{1}$$

Service systems consist of different groups of personnel employed to provide service. For instance, the personnel employed in a clinic include groups of doctors, nurses, laboratory technicians, therapists, pharmacists, etc. If the service cost per unit time, per personnel in a group assigned to a context is S_C, then the total service cost of the context is:

$$Service\,cost = \sum_{j=1}^{m} N_{Pj} S_{Cj} \tag{2}$$

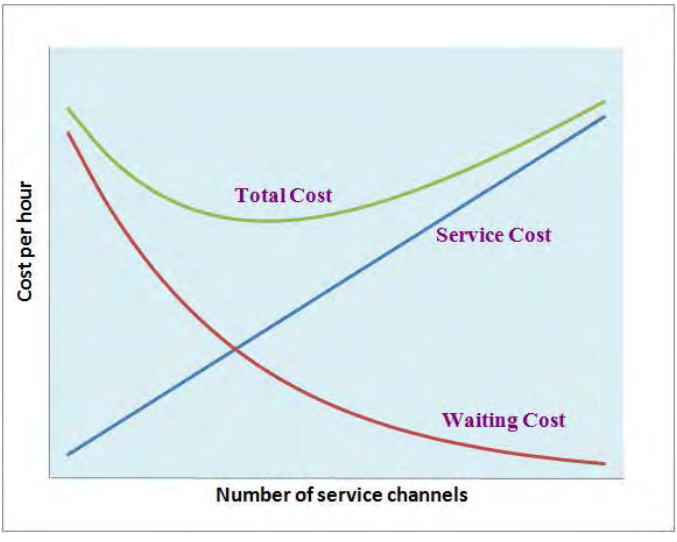


Fig. 2. Waiting cost and service cost in queuing systems

where N_{Pj} is the number of personnel in the j^{th} group, and, m is the number of groups of personnel providing service at the given context. Total cost for each context in the system is:

$$Total\,cost = Q_L W_C + \sum_{j=1}^{m} N_{Pj} S_{Cj} \tag{3}$$

Summing over the costs of all the contexts in the system, the objective function, f becomes:

$$f = \sum_{i=1}^{n} Q_{Li} W_{Ci} + \sum_{i=1}^{n} \sum_{j=1}^{m} N_{Pij} S_{Cij} \tag{4}$$

where n is the total number of contexts in the system.

2.2 Constraints and Penalties

Bounds on Variables

The number of professionals working at each context and the average service time of the context are the decision variables directly related to the cost function. In practical service systems, both these variables have lower as well as upper bounds. The capacity N_P of the server represents the number of personnel or the number of equipment pieces that are assigned to a given context.

$$N_{P<} \leq N_P \leq N_{P>} \tag{5}$$

where $N_{P<}$ and $N_{P>}$ are the lower and the upper bounds, respectively.

Further, each context has an appropriate service time that is usually drawn from an exponential distribution. The service time limits can be expressed as:

$$t_< \leq t \leq t_> \tag{6}$$

where $t_<$ and $t_>$ are the lower and the upper bounds, respectively.

Server Utilization Constraints

In optimization, constraints are usually handled by imposing penalty functions [3, 13]. In stochastic queuing systems, server utilization constraints are laid down by the domain experts' heuristics [7]. Lower server utilization ($\rho < 0.3$) is an indication that the servers are idle. It implies an over-employment of service personnel. The penalty imposed in such a situation is the cost of employing service personnel (i.e., an additional service cost). On the other hand, higher server utilization ($\rho > 0.7$) is an indication that the system is overloaded, leading to potentially long queues in front of the contexts. The penalty imposed in such a case is the additional waiting cost at that particular context. The penalty functions for violating the lower ($\rho < 0.3$) and the upper ($\rho < 0.7$) server utilization constraints are respectively given by:

$$f_{p<} = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{Pij} S_{Cij} \delta_i \tag{7}$$

and

$$f_{p>} = \sum_{i=1}^{n} Q_{Li} W_{Ci} \delta_i \tag{8}$$

where,
$\delta_i = 1$, if the i^{th} constraint is violated, and,
$\delta_i = 0$, if the i^{th} constraint is satisfied.

Customer Satisfaction Constraints

In this section, we discuss the fuzzy service constraints. A model of customer satisfaction with regard to ordering a product is found in [10]. We extend this model of customer satisfaction (CS) towards services.

The membership function μ_S of the service time t is given by:

$$\mu_S(t) = \begin{cases} = 0; & (t \leq t_{min}) \\ = \frac{(t - t_{min})}{(t_{opt} - t_{min})}; & t_{min} < t < t_{opt} \\ = 1; & (t = t_{opt}) \\ = \frac{(t_{max} - t)}{(t_{max} - t_{opt})}; & t_{opt} < t < t_{max} \\ = 0; & (t \geq t_{max}) \end{cases} \tag{9}$$

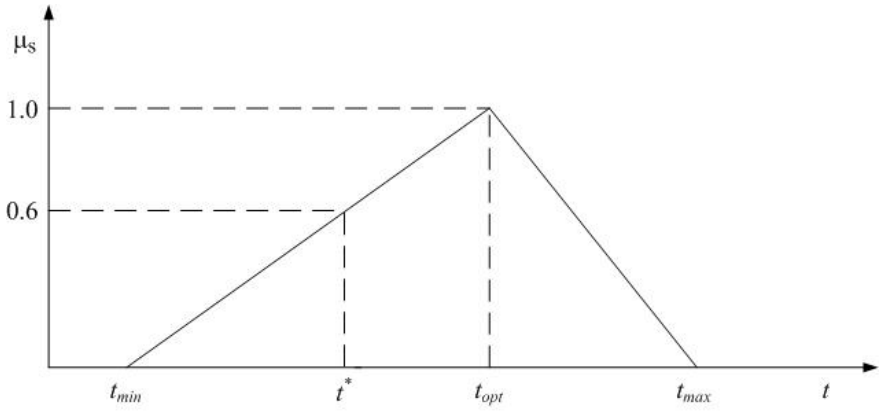


Fig. 3. Fuzzy membership function representing customer satisfaction

The membership function is 0 for $t \leq t_{min}$ and for $t \geq t_{max}$ (Fig. 3). In general, the shorter the service time, the greater is the customer satisfaction. However, in a medical setup, if the time spent by the doctors in treating patients is made very short, the patients feel they have not been listened to, that they have been rushed. On the other hand, if the doctor takes too long in examining the patient, the latter may feel bored. There exists an optimum service time t_{opt} for which the patient is fully satisfied with the medical service ($\mu_S(t) = 1$).

Assuming that the clinic's management policy is to treat the patients so that they have at least 60% service satisfaction, the crisp value of the appropriate service time t^* would be the time corresponding to 0.6 value of the fuzzy membership function, as shown in Fig. 3.

However, if a given context fails to deliver the required level of customer satisfaction, then the waiting cost is imposed as a penalty, since, being dissatisfied with the service, the patient is likely to discontinue the service of the clinic. (Recall that the waiting cost is the profit lost due to a lost customer).

$$f_{CS} = \sum_{i=1}^{n} Q_{Li} W_{Ci} \delta_i \tag{10}$$

Imposing penalties for violating the server utilization constraints ($f_{\rho<}, f_{\rho>}$) and the customer satisfaction constraint (f_{CS}), the objective function with penalties, (f_P) effectively becomes:

$$f_P = f + f_{\rho<} + f_{\rho>} + f_{CS} \tag{11}$$

3 PSO in Service Systems Optimization

3.1 PSO Algorithm

The Particle Swarm Optimization (PSO) algorithm imitates the information sharing process of a flock of birds searching for food. The population-based PSO conducts a search using a population of individuals. The individual in the population is called the particle and the population is called the swarm. The performance of each particle is measured according to a pre-defined fitness function. Particles are assumed to *fly* over the search space in order to find promising regions of the landscape. In the minimization case, such regions possess lower functional values than other regions visited previously. Each particle is treated as a point in a d-dimensional space which adjusts its own *flying* according to its flying experience as well as the flying experience of the other companion particles. By making adjustments to the flying based on the local best (*pbest*) and the global best (*gbest*) found so far, the swarm as a whole converges to the optimum point, or at least to a near-optimal point, in the search space.

The notations used in PSO are as follows: The i^{th} particle of the swarm in iteration t is represented by the d-dimensional vector, $x_i(t) = (x_{i1}, x_{i2}, \ldots, x_{id})$. Each particle also has a position change known as velocity, which for the i^{th} particle in iteration t is $v_i(t) = (v_{i1}, v_{i2}, \ldots, v_{id})$. The best previous position (the position with the best fitness value) of the i^{th} particle is $p_i(t-1) = (p_{i1}, p_{i2}, \ldots, p_{id})$. The best particle in the swarm, i.e., the particle with the smallest function value

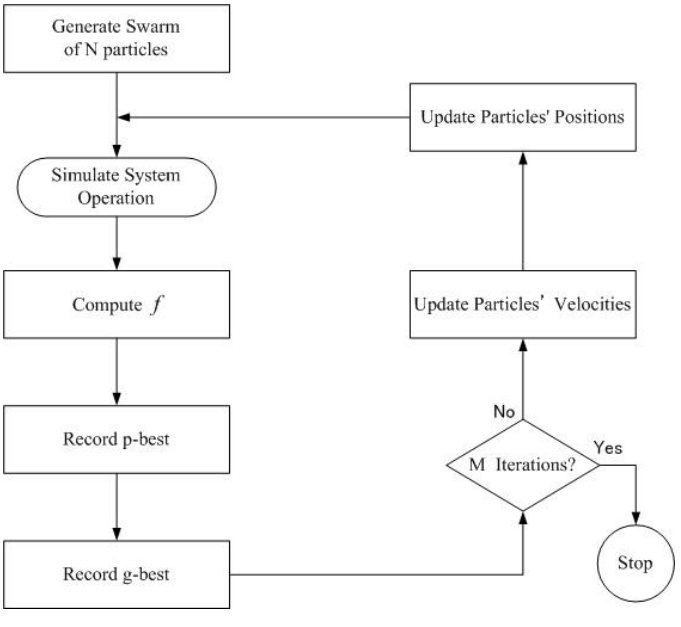


Fig. 4. PSO algorithm

found in all the previous iterations, is denoted by the index g. In a given iteration t, the velocity and position of each particle is updated using the following equations:

$$v_i(t) = wv_i(t-1) + c_1r_1(p_i(t-1) - x_i(t-1)) + c_2r_2(p_g(t-1) - x_i(t-1) \quad (12)$$

and

$$x_i(t) = x_i(t-1) + v_i(t) \quad (13)$$

where, $i = 1, 2, \ldots, NP; t = 1, 2, \ldots, T$. NP is the size of the swarm, and T is the iteration limit; c_1 and c_2 are r_2 are random numbers between 0 and 1; w is inertia weight that controls the impact of the previous history of the velocities on the current velocity, influencing the trade-off between the global and local experiences. A large inertia weight facilitates global exploration (searching new areas), while a small one tends to facilitate local exploration (fine-tuning the current search area). Equation 12 is used to compute a particle's new velocity, based on its previous velocity and the distances from its current position to its local best and to the global best positions. The new velocity is then used to compute the particle's new position. The algorithm is illustrated in Fig. 4.

3.2 Optimization Result

The operational parameters of the small clinic service system are listed in Table 1 and Table 2. These are the inputs to the system simulation. In general, three groups of personnel, Type I (eg. doctors), Type II (eg. senior nurses) and Type III (eg. junior nurses) are assigned to the clinic contexts. The number in each type is bounded by a minimum and a maximum. The service cost per hour per personnel of a given type is also tabulated. Similarly, the service time with its upper and lower bounds for each of the contexts and the waiting cost per hour per customer (patient) are shown in Table 2. The current values of the number of personnel and of the service time that minimize f_P are obtained by the PSO algorithm.

Table 1. Simulation Optimization input(personnel)

Context	Personnel Type 1				Personnel Type 2				Personnel Type 3			
	Number			Cost	Number			Cost	Number			Cost
	Min	Cur	Max	/hour	Min	Cur	Max	/hour	Min	Cur	Max	/hour
Reception	1	1	3	1000	1	1	2	900	0	0	0	800
Diagnosisi1	1	1	3	6000	1	1	3	1200	1	1	2	800
Diagnosisi2	1	1	4	5000	1	1	3	1200	1	2	2	800
Med.Tests	1	1	3	1200	1	1	2	900	0	0	0	800
Prescription	1	1	3	1400	1	1	2	900	1	1	2	800
Physiotherapy	1	1	2	1500	1	1	2	900	1	1	2	800
Accounts	1	1	2	1000	1	1	2	900	0	0	0	800

Table 2. Simulation Optimization input(service time)

Context	Service time per Context (minutes)				Wait cost /customer /hour (yen)
	Min	Cur	Max	Optimum	
Reception	8	12	12	10	1100
Diagnosis1	15	19	30	20	5000
Diagnosis2	16	22	35	22	4500
Med.Tests	15	24	25	19	1200
Prescription	15	15	35	20	1500
Physiotherapy	20	34	55	30	1500
Accounts	3	10	15	6	1200

Table 3. Simulation Optimization output parameters

Contexts	ρ	Q_L	Waiting Cost /day	Service Cost /day	ρ Penalty Cost/day	CS	CS Penalty Cost/day
Reception	0.62	50.00	550000.00	19000.00	0.00	0.00	
Diagnosis1	0.44	0.04	1916.67	80000.00	0.00	0.80	0.00
Diagnosis2	0.30	0.03	1500.00	78000.00	0.00	1.00	0.00
Med.Tests	0.44	0.03	300.00	21000.00	0.00	0.17	
Prescription	0.53	0.29	4325.00	31000.00	0.00	0.00	
Physiotherapy	0.43	0.11	1600.00	24000.00	0.00	0.84	0.00
Accounts	0.37	0.02	260.00	19000.00	0.00	0.56	

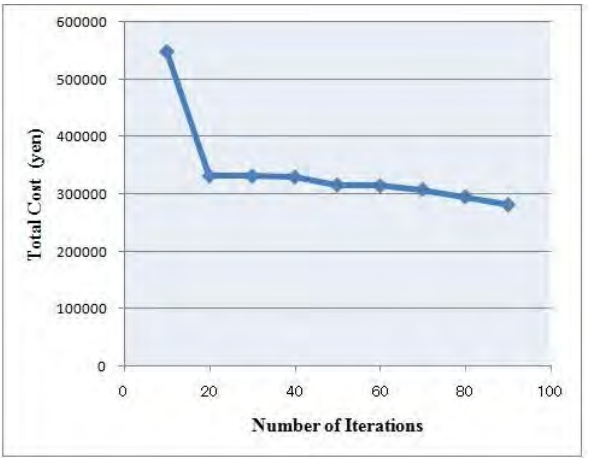


Fig. 5. Convergence of the PSO algorithm

The simulation output parameters are shown in Table 3. Server utilization (ρ), average queue length (Q_L) and customer satisfaction (CS) are the optimized outputs. In the optimized scenario, the server utilization penalties ($f_{\rho<}, f_{\rho>}$) as well as the customer satisfaction penalties (f_{CS}) are all zeroes. For an average inter-arrival time of 5 minutes, the minimum cost of system operation for a day is found to be 281,902 yen.

The search space of our application problem can be estimated as follows. Assume, on an average, that there are 3 personnel in each of the 3 types of groups. Further, assume that the average service time of the contexts in the clinic system ranges from 10 to 30 minutes. Since there are 7 contexts in all, the search space $= 3^7 \times 3^7 \times 3^7 \times 20^7 = 1.34 \times 10^{19}$. Even with a modest number of contexts in the service systems with not so long service time ranges, the search space explodes in size. Hence, the use of the rapidly converging PSO algorithm (Fig. 5) is appropriate.

4 Conclusion

In this paper, we have presented the application of the PSO meta-heuristic algorithm in the optimization of the operation of a practical service system. The cost function is expressed as the sum of the service cost and the waiting cost. Service cost is due to hiring professionals or equipment to provide service to end users. Waiting cost emerges when customers are lost owing to unreasonable amount of waiting for service. Waiting can be reduced by increasing the number of personnel. However, increasing the number of personnel, results in a proportional increase in the service cost. The simulation optimization strategy finds the optimum balance between the two costs so that the overall cost is minimized. PSO obtains the optimum results with rapid convergence even for a very large search space. An extension to this study would be multi-objective optimization.

Acknowledgements

This research has been supported by the Open Research Center Project funds from 'MEXT' of the Japanese Government (2007-20011).

References

1. Anderson, D.R., Sweeney, D.J., Williams, T.A.: An Introduction to Management Science: Quantitative Approaches to Decision Making, 10th edn., Thomson South-Western, Ohio (2003)
2. Banks, J., Carson II., J.S.: Discrete-Event System Simulation. Prentice-Hall, New Jersey (1984)
3. Deb, K.: Optimization for Engineering Design: Algorithms and Examples. Prentice Hall, Delhi (1995)
4. Fishman, G.S.: Principles of Discrete Event Simulation. John Wiley and Sons, New York (1978)

5. Hasegawa, A., Kumagai, S., Itoh, K.: Collaboration Task Analysis by Identifying Multi-Context and Collaborative Linkage. CERA 8(2), 61–71 (2000)
6. Hillier, F.S.: Economic Models for Industrial Waiting Line Problems. Management Science 10(1), 119–130 (1963)
7. Itoh, K., Honiden, S., Sawamura, J., Shida, K.: A Method for Diagnosis and Improvement on Bottleneck of Queuing Network by Qualitative and Quantitative Reasoning. Journal of Artificial Intelligence (Japanese) 5(1), 92–105 (1990)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
9. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
10. McCahon, C.S., Lee, E.S.: Fuzzy job sequencing for a flow shop. European Journal of Operations Research 62, 31–41 (1990)
11. Ozcan, Y.A.: Quantitative Methods in Health Care Management: Techniques and Applications. Jossey-Bass/Wiley, San Francisco (2005)
12. Reeves, C.: Genetic Algorithms. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 55–82. Kluwer Academic Publications, Boston (2003)
13. Smith, A.E., Coit, D.W.: Penalty Functions. In: Baeck, T., Fogel, D., Michalewicz, Z. (eds.) Handbook of Evolutionary Computation, pp. C5.2:1–C5.2:6. Oxford University Press, Oxford (1997)
14. Xu, R., Anagnostopoulos, G.C., Wunsch, D.C.: Multiclass Cancer Classification Using Semisupervised Ellipsoid ARTMAP and Particle Swarm Optimization with Gene Ex-pression Data. IEEE/ACM Trans. Computational Biology and Bioinformatics (TCBB) 4(1), 65–77 (2007)

MS2Web: Applying MDA and SOA to Web Services

Haeng-Kon Kim[1] and Roger Y. Lee[2]

[1] Department of Computer information & Communication Engineering, Catholic Univ. of Daegu, Korea
hangkon@cu.ac.kr
[2] Software Engineering & Information Technology Institute, Central Michigan University, USA
lee1ry@cmich.edu

Summary. Service-oriented architectures (SOA) are touted as the key to business agility, especially when combined with a model-driven approach. Model-Driven Architecture (MDA) is a well-developed concept that fits well with web services, but until now it has been a specialized technique that is beyond practical application scope of most enterprises.

In this paper, we describe the initial investigation in the fields of applying MDA and generative SOA to web services (MS2Web). Our view is that MDA aims at providing a precise framework for generative web service software production. We propose here an initial exploration of some basic artifacts of the MDA and SOA space to web services. Because all these artifacts may be considered as assets for the organization where the MDA is being deployed with SOA, we are going to talk about MDA and SOA abstract components to apply web service business applications. We also discuss the key characteristics of the two modeling architectures, focusing on the classification of models that is embodied by each for web services. The flow of modeling activity is discussed in the two architectures together with a discussion of the support for the modeling flows provided by MDA. We also describe a modeling of case study for web services with two architectures. Our model of the framework — a combined modeling architecture — is introduced which illustrates how the two architectures can be brought together into a synergistic whole, each reinforcing the benefits of the other with case study.

Keywords: SOA (Service Oriented Architecture), MDA(Model Driven Architectures), WSDL(Web Services Description Language), Dynamic Web services, Model Translation.

1 Introduction

Service-oriented architecture (SOA) is an approach to loosely coupled, protocol independent, standards-based distributed computing where software resources available on the network are considered as Services. SOA is believed to become the future enterprise technology solution that promises the agility and flexibility the business users have been looking for by leveraging the integration process through composition of the services spanning multiple enterprises. The software components in a SOA are services based on standard protocols and services in

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 163–180, 2008.
springerlink.com

SOA have minimum amount of interdependencies. Communication infrastructure used within an SOA should be designed to be independent of the underlying protocol layer. Offers coarse-grained business services, as opposed to fine-grained software-oriented function calls and uses service granularity to provide effective composition, encapsulation and management of services. The problems of modeling solutions based on SOA have largely been resolved through the recognition of the importance of loose coupling and the consequent separation of concerns [1, 2]. Reinforced by the Supply-Manage-Consume concept, the separate modeling of solutions and services is a well established practice incorporated into advanced development processes that support SOA, including Select Perspective. Service Interfaces are shared amongst models showing the implementation and reuse of the services [3, 4].

Whilst the use of modeling within SOA is well established, it has suffered from the same issues as modeling in other architectures. The abstraction gap between the level of detail expressed in the model and the level of detail expressed in the code is a key issue. Yet it is the abstraction gap which is one of the key targets for the Model Driven Architecture (MDA). It seems likely, then, that if SOA and MDA can work together they will add value synergistically, leading to greater benefits than either architecture provides in isolation. Yet the two architectures are distant in terms of the way they address the issues surrounding modeling. SOA focuses on the stereotypical roles of models based on separation of concerns. MDA focuses on levels of abstraction, defining the role of models within a process. The question of the compatibility of these two model architectures remains open to solve [5, 6].

Web services represent an evolution of the Web to allow the open and flexible interaction of applications over the Internet. XML Web services are the fundamental building blocks in the move to distributed computing on the Internet.

In this paper, we focus on integrating MDA and SOA component development to enable the development and usage of components by dynamically creating web services for the functionalities of the components. XML Web services standards, which include SOAP, XML, and WSDL, provide a high level of interoperability across platforms, programming languages and applications and have the best aspects of MDA and SOA development. We also describe the initial investigation in the fields of applying MDA and generative SOA to web services (MS2Web). Our view is that MDA aims at providing a precise framework for generative web service software production.

We propose here an initial exploration of some basic artifacts of the MDA and SOA space to web services. Because all these artifacts may be considered as assets for the organization where the MDA is being deployed with SOA, we are going to talk about MDA and SOA abstract components to apply web service business applications. We also discuss the key characteristics of the two modeling architectures, focusing on the classification of models that is embodied by each for web services. The flow of modeling activity is discussed in the two architectures together with a discussion of the support for the modeling flows provided by MDA. We finally describe a modeling of case study for web services

with two architectures. Our model of the framework – a combined modeling architecture – is introduced which illustrates how the two architectures can be brought together into a synergistic whole, each reinforcing the benefits of the other with case study. Our approach makes MDA simpler and more accessible, and improves component re-use. Our model for web services can be interoperable and is the key to success for Model-Driven SOA with achieving business agility through SOA.

2 Related Works

2.1 MDA Frameworks

Model-Driven Architecture (MDA) is currently one of the most exciting approaches for accelerating code development and improving the quality of software in complex systems like embedded systems in ubiquitous era. MDA is an approach to the full lifecycle integration of enterprise systems comprised of software, hardware, humans, and business practices. It provides a systematic framework to understand, design, operate, and evolve all aspects of such enterprise systems, using engineering methods and tools [5, 6, 7]. MDA utilizes models and a generalized idea of architecture standards to address integration of enterprise systems in the face of heterogeneous and evolving technology and business domains. MDA combines computer-aided verification and machine intelligence during modeling to discover and remove design bugs before code reviews and testing. MDA Meta model acts as a filter to extract some relevant aspects from a system and to ignore for all other details. A meta-meta-model defines a language to write meta-models. The application of MDA to a use case begins by focusing on the development of the models. Figure 1 show the MDA frameworks process that includes: Computation Independent Model (CIM): describes concepts of a given domain but does not describe the software system. Platform Independent Model (PIM): describes software behavior that is independent of some platform. Platform Specific Model (PSM): describes software behavior that is specific for some platform. The first step in using MDA is to develop a CIM which describes the concepts for a specific domain. For example, a CIM might describe experiment protocols, or properties of genes. The CIM focuses on the environment and requirements of the system; the details of the structure and processing of the system are hidden or as yet undetermined.

The next step involves developing the PIM. The term “platform” can have various meanings and can include one or more system aspects such as operating system, network configurations, and programming language. The meanings of PIM and PSM models are therefore relative to the definition of platform used in the use case. More important than the definition of platform is the recognition that PIMs and PSMs are supposed to separate aspects of program behavior from aspects of implementation. The third step is developing one or more PSMs which characterize a particular deployment of a software application. This could, for example, focus on the properties of a web application, whether the application should be generated in Java or Visual Basic, or whether the installation was

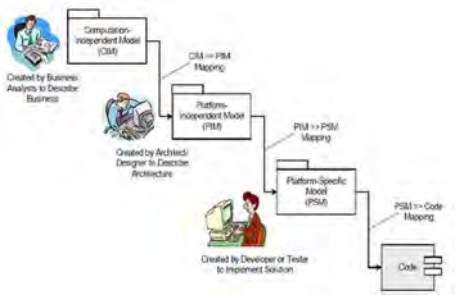


Fig. 1. MDA Frameworks Development Process

for a standalone or networked machine. MDA requires development of explicit transformations that can be used by software tools to convert a more abstract model into a more concrete one. A PIM should be created, and then transformed into one or more PSMs, which then are transformed into code. The mappings between models are meant to be expressed by a series of transformation rules expressed in a formal modeling language. A CIM is a software independent model used to describe a business system. Certain parts of a CIM may be supported by software systems, but the CIM itself remains software independent. Automatic derivation of PIMs from a CIM is not possible, because the choices of what pieces of a CIM are to be supported by a software system are always human. For each system supporting part of a CIM, a PIM needs to be developed first.

It is possible for concepts defined in a CIM to be automatically associated with properties defined in a PIM. For example, the concept protein defined in a CIM about proteomics experiments could be associated with PIM concepts such as a help feature that defined protein for users or a drop down list of protein names [4, 5].

2.2 Developing the Service Oriented Architecture

MDA involves developing models which lead to automatically generated applications. These applications will possess generic features that need to be customized

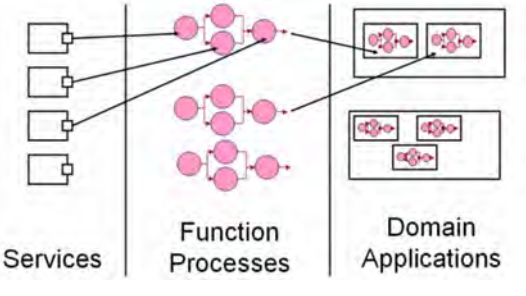


Fig. 2. SOA Three Architectural Perspectives

to suit specific domains. To support specialized functionality, MDA advocates the development of a Service Oriented Architecture (SOA). The architecture comprises a collection of services which implement well defined interfaces and interact with the other parts of the generated application. Service-Oriented Architecture (SOA) is an architectural framework that supports integrating business tasks as linked services that can be accessed when needed over a network. Within SOA architecture, all functions, such as check service inventory, software distribution, and payment services etc., are defined as services. For SOA there are three important architectural perspectives as shown in Figure 2 [8].

- *The Application Architecture*

This is the business facing solution which consumes services from one or more providers and integrates them into the business processes.

- *The Service Architecture*

This provides a bridge between the implementations and the consuming applications, creating a logical view of sets of services which are available for use, invoked by a common interface and management architecture.

- *The Component Architecture*

This describes the various environments supporting the implemented applications, the business objects and their implementations.

SOA Architectural roles can be reflected using UML stereotypes. Strong architectural roles are assigned to the executable artifacts deployed by projects delivering into the SOA. In SOA, each executable artifact is the implementation of a solution, one or more services or part of the technical architecture. Any of these items may invoke services offered by other deployed executable artifacts. A clear separation of concerns in this way reflects a key characteristic of SOA – the decoupling of different layers of the application architecture.

Each deployed artifact also experiences its own life cycle – it evolves, develops and is maintained independently of other deployed artifacts. The loosely coupled nature of the Service Oriented Architecture avoids the "brittle interface" problem and encourages version independence between deployed configuration items and allows them to evolve (relatively) independently. Consequently, each model that is an abstraction of an implementation item represents either a component or a solution. These models can be stereotyped as component or solution models.

2.3 Web Services Architecture

Web services are self-contained, modular applications that may be described, published, located, and invoked over a network, generally over the Web [9, 10]. They are rapidly emerging as important building blocks for business integration.

They are also finding important applications in business-to-business, business-to-consumer, and enterprise application integration solutions [11]. Important factor in this approach is the independence of interactions from the platform, programming language, middleware, and implementation of the applications involved. The Web Services architecture is the logical evolution of object-oriented analysis and design, and the logical evolution of components geared towards the architecture, design, implementation, and deployment of e-business solutions, both approaches have been proven in dealing with the complexity of large systems. Several essential activities need to happen in any services-oriented environment:

- A Web service needs to be creating, and its interfaces and invocation methods must be defined.
- A Web service needs to be published to one or more intranet or internet repositories for potential user to locate.
- A Web service needs to be located so that potential users may invoke it.
- A Web service may need to be unpublished when it is no longer available or needed.

Basically Web Services comprise of three main components:

- *Service:* Processing an XML document that it receives through some combination of transport and application protocols.
- *XML document:* This is the keystone of any web service because it defines all the application specific information that a service consumer sends to the service for processing.
- *Address:* This component is also called port reference, which is a protocol binding combined with a network address that a requester may use to access the service. This identifies the location of the service using a protocol (for example: TCP or HTTP)

Web Services architecture requires three fundamental operations: Publish, find, and bind. Service providers publish services to a service broker. Service requesters find required services using a service broker and bind to them. These ideas are shown in Figure 3.

A Web service is an interface that defines a collection of operations that are network-accessible through a standardized XML messaging. A Web service

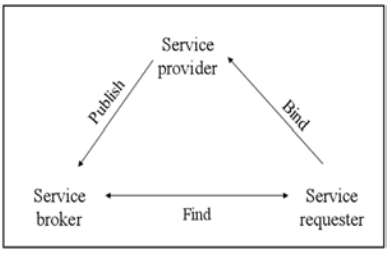


Fig. 3. Publish, find, and bind

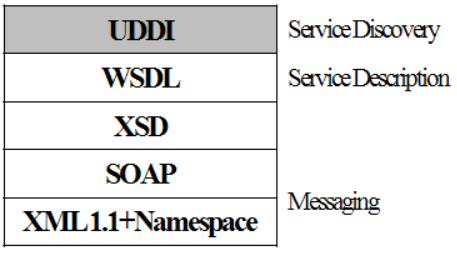


Fig. 4. Web Services technologies

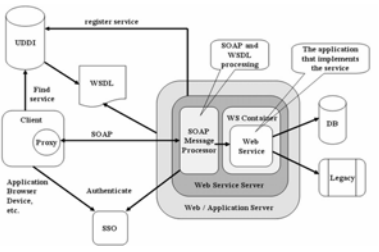


Fig. 5. Web service run time environment

performs a specific task or a set of tasks. It is described using a standard, formal XML notation called its service description that provides all the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location [12].

Web services are a collection of technologies that defines the following specifications as in figure 4:

- Interface to a UDDI (Universal Description Discovery and Integration)
- A way of describing services, WSDL (Web Services Description Language)
- A protocol for talking to the services, SOAP
- Communication format, XML (Extensible Markup Language)

The interface of a web service is described in an XML format called the *WSDL*. A WSDL file contains descriptions of one or more interfaces and binding information for one or more services. *UDDI* is a public or private registry of web services. SOAP protocol consists of two parts, enquiry and publisher may gain access to modify and delete information. UDDI builds on SOAP and WSDL. The runtime server listens for SOAP requests from the client. Upon receiving the request, the server then determines how to handle the message that is associated with the request during run time as in figure 5.

3 Design MS2Web (MDA and SOA to Web services) Frameworks

Model-Driven Architecture (MDA) is a well-developed concept that fits well with SOA, but until now it has been a specialized technique that is beyond the scope of most enterprises. Service-oriented architectures (SOA) are touted as the key to business agility, especially when combined with a model-driven approach. The practical benefits of SOA are being increasingly recognized in many industries and applications of Information Technology, especially to Web related application. SOA eliminates the traditional "spaghetti" architecture that requires many interconnected systems to solve a single problem. Rusty adds that SOA provides a bridge between the business and technical worlds, enabling customers to talk about what they really need. But understanding requirements is just the first

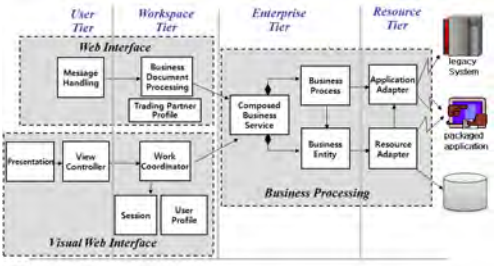


Fig. 6. Architecture and Procee for MS2Web

step towards business agility. In a truly agile enterprise, the IT infrastructure is aligned with the business structure and it can easily be adapted to meet business needs. The promise of SOA is that this can be achieved by business people who can configure and orchestrate the SOA services. This implies that the business people have a model of the enterprise services, that they configure the model to reflect business needs, and that this translates directly to the IT implementation, ideally automatically. Such a model-driven approach is well suited to SOA, because the basic architectural building blocks are services, which can be described in this paper as the Web Services Definition Language (WSDL). Figure 6 shows the architecture and proceeds for applying web service application with MS2Web in this paper. Combining a service-oriented modeling architecture with MDA for web services can bring many unique benefits. Firstly the clear organization of models and information based on the stereotypes derived from the service-oriented architecture and select perspective as development process. Secondly the productivity, quality and impact analysis benefits of the use of MDA with its emphasis on automation, transformation and synchronization. MS2Web solution for MDA in our approach is uniquely positioned to take advantage of the unified modeling architecture which results from bringing these two key architectures together.

MS2Web for MDA combines a uniquely powerful implementation of the web services vision, together with the industry leading solutions for modeling service-based solutions.

3.1 Combining SOA and MDA with Components

The MDA component of the product in our approach MS2Web features all of the key features of MDA – three levels of model abstraction, the ability to mark and transform and to synchronize related models when they have been created and combine with SOA. The product can be extended with new patterns which meet customer's unique architectural needs. The MDA organization for SOA and web services may be viewed as a set of artifacts, some being standard building blocks, some being user developed. We may envision, in the not too far future, an organization starting with a hierarchical library of meta-models and extending it as an adaptation to its own local context (models as assets). Model reusability in

MS2Webservices will subsume code reusability, with much more efficiency. This may be seen as orthogonal to code class libraries (e.g. Java, Swing, EJB, etc.). Inside a company, the various business and service models will be developed and maintained to reflect the current situation based on our model. The purpose of the MS2Web will consist in making it more general and more universal. The UML Meta model in MS2Web plays different roles in the MDA. First it defines the language used for describing object-oriented software artifacts. Second, its kernel is synchronized with the MOF for practical reasons as previously mentioned. There is much less meta-modelers than modelers (people building models). As a consequence it is not realistic to build specific web services for the first category of people. By making the MOF correspond to a subset of UML, it is possible with some care to use the same tool for both usages. As a consequence the MDA is not only populated by first class MOF meta-models, but also with UML dialects defined by UML profiles for specific purposes languages. This is mainly done for practicality (widening the market of UML tools vendors) and there is some redundancy between UML profiles and MOF metamodels.

In the case web service with SOA development becomes popular or widely available. The CWM (Common Warehouse Metadata) provides the means to partially deal with legacy systems (relational databases and data warehouses). This is an interesting example of a complex meta-model structured as a set of modules dealing with various concerns like object modeling, data modeling, data transformations, business information, type mapping, data warehousing, and data mining. This example shows how the notion of composite model is becoming important. One well-known distinction on meta-models is between products and processes.

A static model is invariant while a dynamic model changes over time. This should not be confused with the static and dynamic aspects of systems. The most common situation is a static model of a dynamic system in web services. The fact that a model is executable or non-executable has nothing to do with the previous property. A Java program may be naturally considered as an executable model. Unless extended, a UML model is non-executable. The way to give some specific executability semantic to a UML model is by using the "Action Semantics" meta-model.

Like any software component, MS2Web components in our model may be atomic or composite with SOA to apply it to web service application. An atomic model of MS2Web contains only basic elements. A composite model contains at least another model. The possibility of composition is defined at the meta-model level. The containment hierarchy for models is distinct from the specialization hierarchy. Our model may be primitive or derived. A derived model may be obtained from other models (primitive or derived) by a derivation operation or a sequence of such operations. A derivation operation is a simple and deterministic model transformation. Our model may contain functional or non-functional elements to combine with SOA. Typical non-functional elements are related to QoS properties like performance, reliability, security, confidentiality, etc. The elements of a non-functional model are usually related to specific elements in a base

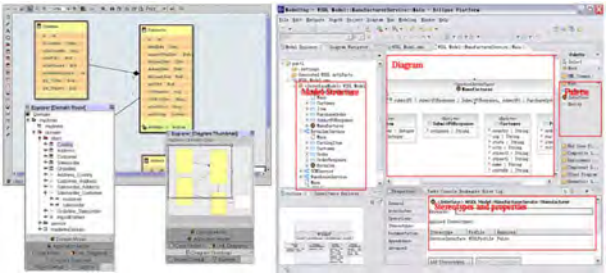


Fig. 7. Combining MDA and SOA for MS2Web with Components

functional model. A more general solution consists in using a correspondence model to state the explicit associations between non-functional and functional elements.

An essential MS2Web model is a model that is intended to stay permanently in the model our repository system. A transient model is disposable and it has been used for some temporary purpose and has not to be saved.When we have a conversion to be done between an important numbers of different meta-models between web servers, it may be interesting to define a pivot meta-model to facilitate the transformation process. UML is a product meta-model for object-oriented software artifacts. We may also consider models of legacy systems.

One important kind of MS2Web model that is being considered now is the correspondence model between MDA and SOA as in figure 7.

A correspondence model explicitly defines various correspondences that may hold between several models. In the usual case, there are only two models: the source and the target.

There may be several correspondences between a couple of elements from source and target. The correspondences are not always between couples of elements and they are strongly typed. There is not yet a global consistent view on correspondence models since this problem is appearing from different perspectives with web server. The traceability a commonly found model that we should not ignore is a source program, written in a given programming language. The metamodel corresponds to the formal grammar of the language and the model is executable. The same source program could be in turn considered as a system and other models can be extracted from it (static analysis). One particular execution of this program may also be considered as a system and other models can be extracted from this execution. A typical example is an execution trace that could be considered as a model, based on a simple specific metamodel. The concept of "just-in time" model production is presently taking shape. All models generated by a given execution program should not be based on the same meta-model.

To consider how these models can relate to one another, consider a system that is intended to manage sales transactions. The CIM could include the definitions of a customer, a product, an employee or a sales request. These concepts would

have the same properties whether they were being represented in an electronic form, a database or a print out. In this sense, "customer" and "product" are defined in ways that are independent of the way they are computed in any part of the software system. The PIM and PSM models in MS2Web would be developed depending on the intended purpose of the auto-generated software.

Applying MDA begins with deciding the scope of functionality of web server software to be generated. The next step is to identify generic aspects of coding which would be reflected by tedious repetitive programming. For example, if adding a new form field requires creating another text field, another label and the same validation logic, it is a task that becomes generic enough to consider for software generation activities. Features which show a high degree of variance in coding are best encapsulated by services that form part of the SOA. The most important part of developing the models is to determine what properties are common to all deployments. This helps inform the process of crafting the PIM and PSM models. The main MDA models and the mappings which relate them are supposed to be expressed in a standard modeling language such as UML. This allows developers to use MDA tools that generate the generic aspects of the applications. Usually the finished product will not be specialized enough to handle all needs of the business use case. Therefore, domain-specific services are developed which can be linked to parts of the auto-generated applications.

3.2 Mapping MDA with SOA to Web Service

MDA involves developing models which lead to automatically generated applications. These applications will possess generic features that need to be customized to suit specific domains. To support specialized functionality, MDA advocates the development of a Service Oriented Architecture (SOA). Figure 8 shows the mapping architecture for MS2Web. The architecture comprises a collection of services which implement well defined interfaces and interact with the other parts of the generated application. The application can be just a screen, which offers a complete set of functions, but behind the screen, calls other web services to do the work. The user may also dynamically change the functionality of the

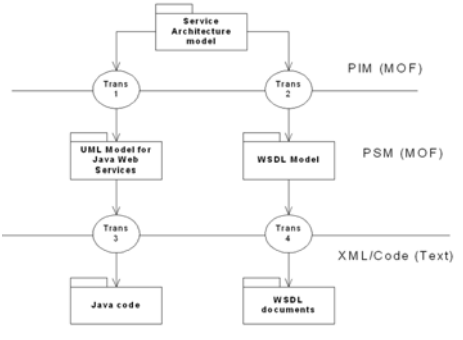


Fig. 8. Mapping Architecture for MS2Web

application, and the application would adapt to the new requirements and call the corresponding services.

A transformation t transforms a MDA and SOA model *MSa* into Web server model *Wb* as:

$$MS2Webt : MSa \rightarrow Wb$$

Model *Ma* is supposed based on meta-model *MMa* and model *Mb* is supposed based on web meta-model *MMweb* as:

$$MS2websem(MSa, MMa)\ MS2\ websem(Wb, MMweb)$$

As a matter of fact, a transformation is like any other model. So we'll talk about the transformation model as:

$$MS2Webt : Mt : MSa \rightarrow Wb$$

Obviously since *Mt* is a model, we postulate the existence of a generic transformation meta-model *MMt*, which would be similar to any other MOF based MDA meta-model as:

$$MS2websem(Mt, MMt)MS2websem(MMt, MOF)MS2websem(MOF, MOF)$$

The nice property here is that we may envision the possibility of producing transformations

$$(Higherorderfunctions) : MS2WebMx : Mu \rightarrow Mt$$

The existence postulate for *MMt* is based on the forthcoming result of the transformation language.

We may also understand the need to check before applying a transformation the consistency of this transformation by studying the relation between both meta-models. Furthermore there is a subtle and important aspect that is related to transformation, which is traceability. A transformation operation should produce a web traceability model as; *Wtr* between the elements of the target model *Wb* and the elements of the source model *Ma*. Obviously this traceability model is based on a given meta-model *MMtr*. There are obvious relations between *MMa*, *MMweb*, *MMt* and *Wtr*.

The study of these relations is at the heart of the definition of the MDA. Presently the main issue in the MDA initiative is to succeed in defining "transformations as models". This means that there should be a generic MOF-compliant transformation meta-model defining some kind of UTL (Unified Transformation Language).

Exploitation on these dynamically generated models may be done concurrently with the execution of the program. There is currently a lot of thinking about platform description models (PDM). Very often these are kept implicit in the MDA presentation. This is in contradiction with the global goal of the MDA that is to make explicit the notion of platform independence and the notion of platform binding. The concept that comes closer to the idea of a platform is the

notion of virtual machine. A virtual machine may be built on top of other virtual machines and the relations of abstraction between different virtual machines may be made explicit.

We need a classification of various platforms according to the technology they are using or to the underlying paradigm on which they are based: objects, components, services, rules, services, transactions, etc. Practically we need models of most popular platforms like CORBA, EJBs, UNIX, Windows, etc. The CCM may be considered to contain such a material while proprietary or collective efforts are producing other models (like the Sun Java Community Process). Here also a platform model may be composite in the sense that it may contain several different components. Many efforts are presently aiming at capturing a model of the web as a specific MDA platform, in order to be able to automatically generate systems for this important target.

When the notion of PDM and virtual machine is clarified we may then tackle the definition of a PIM, a model containing no elements associated to a given platform. In other times this was simply called a business model, but as for platform models we need to progress now towards a less naive and a more explicit view. The first idea is that the PIM is not equivalent to a model of the problem. Many elements of the solution may be incorporated in a PIM as long as they don't refer to a specific deployment platform. For example we may take algorithms hints into account. There may be several PIM to PIM transformation before arriving to the state where a PIM may be transformed into a PSM. To express this idea, some use the notion of a CIM (Computation Independent Model), which is a PIM where the problem has not yet been worked out as a solution.

We could mention for example test models, the way they are produced and the way they may be used and many more categories of models and metamodels. We may then attempt now a classification of operations on models and meta-models. Some operations apply on a single model and are called monadic by opposition to dyadic operations applying to two models. Operations applying on MDA and SOA models are rarer. A model may be built, updated, displayed, queried, stored, retrieved, serialized, etc. When most of these operations are applied, there is an implicit or explicit meta-model for both MDA and SOA. A MDA supporting tool has very often an integrated built-in UML metamodel. Sometimes it is possible for such a case tool of dynamically adapt to new versions of the UML meta-model. In a similar way, a meta-model may be built, updated, displayed, queried, stored, retrieved, serialized, etc. Efficiently storing and u a model or a meta-model to/from persistent storage is not always easy to implement, especially when configuration and version management are involved. In many cases using simple file systems after XMI serialization lacks efficiency and don't scale up. The important question here is how the view is specified and if this operation may be considered as a dyadic operation producing another model. There are many other apparently monadic operations that turn out to be dyadic, if we are able to define the argument as a meta-model or a model. Some examples are measure, verification, normalization, optimization, etc.

Rapid prototyping is a special kind of operation that associate some limited executability to a model in order to interactively evaluates its properties before transforming it into executable systems. This may not be done with any meta-model. A typical usage is to find design errors in an initial UML model for example. Obviously one of the most popular operations on models is code generation. One may convert a UML model into a Java or a C++ programs. Many aspects of this kind of operation should be considered here. In some cases we may consider bidirectionality, i.e. backward as well as forward transformation, with for example transformations of C# code into UML code. However one should not be misled by the apparent simplicity of these transformations.

Usually the underlying algorithms are quite simple and much progress will be made in the coming years in this area. If we consider the target language (C++, C#, Java, etc.) as defined by a meta-model (corresponding to its grammar), then we may envision generic transformations and really parameterized tools. Many dyadic operations on models should also be discussed. A confrontation of two models may produce the similarities and differences between two models, possibly as a new model. The relations between the involved meta-models are obviously of higher importance. An alignment of two models may allow to define a new model if some equivalence rules would have been defined. A fusion/merge of two models is more complex operation since it supposed the the weaving rules have been specified in a third model. There are many other notions to discuss, for example the life cycle of models and meta-models. When describing the life cycle of a model, we can make the distinction between incremental verification and batch verification. When considering the life cycle of a meta-model for example we should also take into account its evolution after being put in service. The need to cope with various version numbers of the UML meta-model is an example of how serious this problem may be. Obviously the most apparent components in an MDA workbench are the precise tools composing our workbench. Fortunately in this context we should be able to propose a rather precise definition of a tool: it is an operational implementation of a set of operations applicable on specific models. The meta-models supported by a tool should be exhaustively and explicitly defined.

4 Frameworks Modeling and Implementation

4.1 Dynamic Web Service Generation with MS2Web

To dynamically generate a web service with MS2Web component requires knowledge on the language and the platform of the component at the PSM layer. There are various tools and technologies available to create web services. The component functionalities may be converted into a web service based on the language and features of the language in which the component is implemented. Dynamic generation of web service with MS2Web depends on the nature of the component and the appropriate tool or technology must be used to generate a web service dynamically.

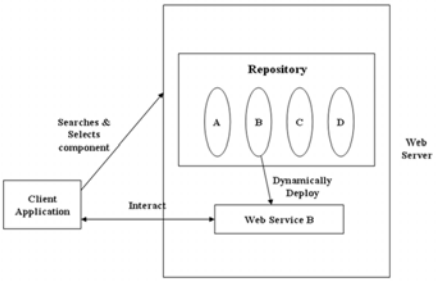


Fig. 9. Dynamic Web Service generation

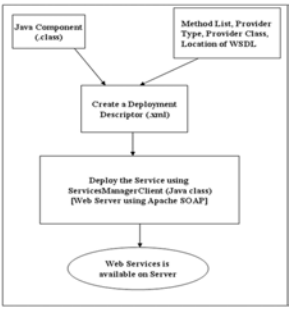


Fig. 10. Web Services from Java Components

A web service is to be dynamically created for the component desired by the client as shown in Figure 9. When the component is selected, the main web service can create a dynamic web service using the component DLL. Then the address or WSDL file of this newly created web service can be given back to the client. When the client is done with the method calls, it can either send message to the consumer web service or the dynamic web service to terminate the web service. Thus the web service is created, used and then deleted as and when required by the client.

This would take the load off the consumer web service. Web services can be developed in any language and used by programs written in any language or platform. All the client needs is a valid WSDL file. Figure 10 shows the sample web service from the Java components at the PSM layer of MS2Web.

4.2 Implementation Web Service with MS2Web

The implementation of dynamically deploying Java components on a web server with MS2web has been effectively completed in this research using the features of Apache SOAP. A deployment descriptor is created using Apache SOAP for the component to be converted into a web service. This is an XML file that describes the component functionality as a web service. The service is deployed

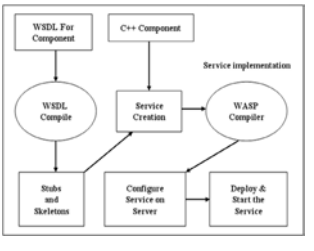


Fig. 11. Web Services from C++ Components

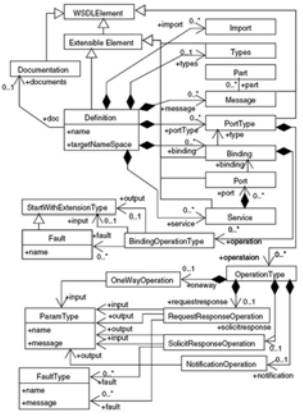


Fig. 12. Web Services from C++ Components

Fig. 13. MS2Web related Static Modeling

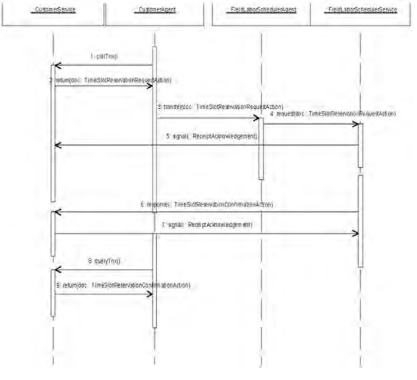


Fig. 14. MS2Web related dynamic Modeling

on the web server using Apache SOAP. Implementation is also possible using the WASP Server, which is a platform-independent, easy-to-use, high performance Web services runtime environment for creating, deploying and managing Web services in Java and J2EE applications, WASP Server offers the industry, best performance, interoperability, reliability and security. It is also possible to create web services for C++ components using the WASP Server for C++. As in figure 11 it is the most advanced, standards-compliant development and deployment environment for creating and consuming secure Web services in C and C++ applications. WASP Server guaranteed to interoperate with a broad range of SOAP implementations, including Microsoft .NET, J2EE servers, Apache AXIS, and others. The steps in the conversion of a C++ component into a web service using the WASP toolkit with MS2Web is shown in figure 12.

A WSDL Compiler generates the stubs and skeletons are used in the creation of a web service. The service is implemented using the WASP compiler and configured to be deployed on the web server.

We develop a MS2Web server prototyping as in figure 13 and 14 from the modeling environment for the conversion from a WSDL document attached to a MS2Web model. Figure 13 shows the corresponding view of the UML model for a WSDL document. For clarity, the WSDL document is simplified by leaving out a few elements and attributes, as well as by removing all of the XML namespace information. Figure 14 shows several UML elements modeling concrete WSDL document elements. The arrows link these elements with corresponding representation in the WSDL model structure.

5 Conclusions and Future Work

MDA is about an important paradigm change, from objects to models, from interpretative to generative approaches. However, with a lot of hype accompanying the model engineering movement, it seems important to clearly state the essence of this new way of designing information systems and conducting web service engineering projects.

In this paper, we focus on integrating MDA and SOA component development to enable the development and usage of components by dynamically creating web services for the functionalities of the components. XML Web services standards, which include SOAP, XML, and WSDL, provide a high level of interoperability across platforms, programming languages and applications and have the best aspects of MDA and SOA development. We also describe the initial investigation in the fields of applying MDA and generative SOA to web services (MS2Web). Our view is that MDA aims at providing a precise framework for generative web service software production. We also describe a modeling of case study for web services with two architectures. Our model of the framework – a combined modeling architecture – is introduced which illustrates how the two architectures can be brought together into a synergistic whole, each reinforcing the benefits of the other with case study. Our approach makes MDA simpler and more accessible, and improves component re-use. Our model for web services can be interoperable and is the key to success for Model-Driven SOA with achieving business agility through SOA. In the future, we plan to construct MS2Web repository to reuse the related assets with it. We also concern about the practical case studies and verification our approaches.

References

1. Smith, M., Friese, T., Freisleben, B.: Model Driven Development of Service-Oriented Grid Applications. In: International Conference on Internet and Web Applications and Services/Advanced International Conference on AICT-ICIW apos 2006, vol. 19(25), pp. 139–149 (2006)
2. Radhakrishnan, R., Wookey, M.: Model Driven Architecture Enabling Service Oriented Architectures. In: Whitepaper SUN Microsystems, pp. 1–13 (2004)

3. Smith, M., Friese, T., Freisleben, B.: Towards a Service-Oriented Ad Hoc Grid. In: Proceeedings of the 3rd International Symposium on Parallel and Distributed Computing, Cork, Ireland, pp. 201–209. IEEE Press, Los Alamitos (2004)
4. Skogan, D., Gronmo, R., Solheim, I.: Web Service Composition in UML. In: Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conference, pp. 111–120 (2004)
5. Bezivin, J., Gerbe, O.: Towards a Precise Definition of the OMG/MDA Framework ASE 2001, San Diego, USA (2001)
6. D'souza, D.: Model-Driven Architecture and Integration: Opportunities and Challenges Version 1.1 (2001), ftp://ftp.omg.org/pub/docs/ab/01-03-02.pdf
7. Object Management Group Meta Object Facility (MOF) Specification. OMG document (1997), http://www.omg.org/technology/documents/formal/mof.htm
8. Object Management Group MOF 2.0 Query/Views/Transformations RFP. OMG document (2004), http://www.omg.org/docs/ad/04-04-01.pdf
9. Object Management Group Model-Driven Architecture. OMG document (2000), http://www.omg.org/mda/specs.htm
10. Frankel, D.: Using Model-Driven Architecture to Develop Web Services. IONA Technologies PLC, p. 4 (2002)
11. Fraternali, P., Paolini, P.: Model-Driven Development of Web Applications: The Autoweb System. ACM Transactions on Information Systems 28, 323–382 (2000)
12. Mulye, R.: Modeling Web Services using UML/MDA (2005), http://lsdis.cs.uga.edu/ranjit/academic/essay.pdf
13. OASIS Web Services Resource Framework (2004), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

Security Analysis of One Verifiable Encryption Scheme

Lihua Liu[1] and Zhengjun Cao[2]

[1] Department of Mathematics, Shanghai Maritime University, China
lhliu@yahoo.cn
[2] Department of Mathematics, Shanghai University, China
Computer Sciences Department, Universite Libre de Bruxelles, Belgium
caozhj@yahoo.cn

Summary. In 1998, W. Mao proposed a verifiable encryption scheme. In the scheme Alice shall encrypt two prime numbers P and Q and disclose $N = PQ$. Bob shall verify the correctness of the encryption under an agreed public key. In the short paper, we show that Alice can only disclose $N = PQ \bmod q$, where q is the order of the cryptographic group used for zero-knowledge proof. Actually, the proof of bit-length proposed can only show the bit-length of the residue $\hat{P} \in \mathcal{Z}_q$ in stead of $P \in \mathcal{Z}$. To fix the scheme, it's sure that the order of the cryptographic group should be *unknown* by the prover. That means we should introduce another RSA modulus and base the Mao's scheme on RSA setting instead of the original ElGamal setting.

1 Introduction

The integer factorization problem forms an important class of protocols such as RSA [16], Rabin [15], Fiat-Shamir [10] and Guillou-Quisquater [11]. The factorization of an integer N (e.g., an RSA modulus) can be fairly shared. Here, fairness means to share a secret among a plural number of participants called shareholders. This can be achieved via verifiable secret sharing (VSS, e.g., [1, 13, 14]). VSS schemes are multi-party protocols which require the shareholders stay on-line to receive and verify messages. They are very inefficient, and give the user little freedom to choose shareholders they trust. We remark that such verifiable encryptions are intensively used in some cryptographic schemes, such as electronic cash systems, group signatures [7], publicly verifiable secret sharing schemes [3], and other zero-knowledge protocols [8, 9].

In 1998, W. Mao [12] proposed a publicly verifiable secret sharing scheme for establishing correct sharing of the factorization of an integer. The public verifiability means that the secret sharing need not use a plural number of on-line participating verifiers. A single verifier will suffice for the verification job, and because the verification will only use public data such as the public keys of the un-participating shareholders, the verifier can be anybody and the verification can be repeated. Thus, Mao's scheme achieves a guaranteed correctness in secret sharing.

In Mao's protocol Alice shall encrypt two prime numbers P and Q and disclose $N = PQ$. Bob shall verify the correctness of the encryption under an agreed

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 181–187, 2008.
springerlink.com

public key. In the short note, we show that Alice can only disclose $N = PQ \bmod q$, where q is the order of the cryptographic group used for zero-knowledge proof of knowledge of a discrete logarithm. Actually, the proof of bit-length proposed can only show the bit-length of the residue $\hat{P} \in \mathcal{Z}_q$ in stead of $P \in \mathcal{Z}$. Hence the premise in the key lemma in Mao's scheme cannot be satisfied. Bob cannot be convinced that $N = PQ$. Therefore, the whole protocol fails. To fix the scheme, it's sure that the order of the cryptographic group should be *unknown* by the prover. That means we should introduce another RSA modulus and base the Mao's scheme on RSA setting instead of the original ElGamal setting. Thus, an increase in the cost of modified Mao's scheme definitely occurs.

2 Review

2.1 Notation and System Setup

Let $\mathcal{Z}_p$ denote the ring of integers modulo p and $\mathcal{Z}_p^*$ denote the multiplicative group modulo p. For integers a, b, we write $a \mid b$ if a divides b and $b \neq 0 \bmod a$ if otherwise. Let $|a|$ denote the bit length of a, and *abs(a)*, the absolute value of a. Let r be a large prime such that $q = 2r + 1$ and $p = kq + 1$ are also prime where k is an even number. Let $h \in \mathcal{Z}_q^*, g \in \mathcal{Z}_p^*$ be elements of order r and q, respectively, and set $H =< h >, G =< g >$ with multiplication as the respective group operations. The system will also setup a fixed element $f \in G$ such that nobody knows $\log_g(f)$. Such an element can be chosen by a trusted center.

2.2 Protocol: Verifiable Encryption of Integer Factorization

Task. In this protocol Alice shall encrypt two prime numbers P and Q and disclose $N = PQ$ to Bob. Bob shall verify the correctness of the encryption under an agreed public key.

Data preparation by Alice. Generates two primes $P, Q, abs(|P| - |Q|) < C$, where C is a pre-specified small constant (in practice, $C \leq 20$). Computes $V_1 = g^P \bmod p, V_2 = g^Q \bmod p$, and $N = PQ$. Encrypts P in A_1, B_1, and Q in A_2, B_2 as follows:

$$(A_1 = h^{K_1}, B_1 = Y^{-K_1} P) \pmod q,$$
$$(A_2 = h^{K_2}, B_2 = Y^{-K_2} Q) \pmod q$$

where $K_1, K_2 \in_R \mathcal{Z}_r$. $Y = h^X \bmod q$ is an agreed public key. Its discrete logarithm X has been fairly shared among a number of shareholders. The encryption will use the ElGammal cryptosystems. It can be described as follows.

1. Alice sends to Bob: $A_1, B_1, V_1, A_2, B_2, V_2, (K_1 + K_2) (\bmod r)$ and N.
2. Bob verifies that $(h^{K_1+K_2} = A_1 A_2,\ N = Y^{K_1+K_2} B_1 B_2) \bmod q$.
3. Alice shows to Bob evidence that N consists of only two distinct primes.
   3-1. Alice proves to Bob that (A_1, B_1) encrypts $\log_g(V_1)$, and (A_2, B_2) encrypts $\log_g(V_2)$, and the encryptions are under the agreed public key Y.

3-2. Alice shows to Bob $|P|$ and $|Q|$ using a sub-protocol to be described in the next subsection. He verifies that $|P|+|Q| \leq |N|+1$ and $abs(|P|-|Q|) < C$.
4. Bob accepts the proof if every checking in the above passes, otherwise rejects.

Upon successful termination of a run, Bob will certify N as Alice's public key and archive the $A_1, B_1, V_1, A_2, B_2, V_2$ for possible future recovery of P and Q.

2.3 Proof of Bit-Length $|P|$ and $|Q|$

Let $m = |x| - 1$ and

$$x = a_0 2^0 + a_1 2^1 + \cdots + a_m 2^m \tag{1}$$

for $a_i \in \{0, 1\}$ and $i = 0, 1, \cdots, m$, be the binary presentation of x. Alice want to convince Bob that the bit-length of the committed value x is $m+1$. They can proceed as follows.

(1) **Part-I**

1. Alice computes $V_1 = g^x \bmod p$, and convinces Bob that she knows $\log_g V_1$ using any Zero-Knowledge Proof.
2. Alice chooses $u_0, u_1, \cdots, u_m \in_R \mathcal{Z}_q$. She computes
$$u = u_0 2^0 + u_1 2^1 + \cdots + u_m 2^m \mod q \tag{2}$$
and $E_i = E(a_i, u_i) = g^{a_i} f^{u_i} \bmod p$, for $i = 0, 1, \cdots, m$.
3. Alice sends E_i and u to Bob.
4. Bob checks whether $V_1 f^u \stackrel{?}{=} \prod_{i=0}^{m} E_i^{2^i} \mod p$.

(2) **Part-II**

Common input: $E, f, g \in G$
Prover's input: $z \in \mathcal{Z}_q$
To prove either $E = f^z$ or $E = gf^z \pmod p$

1. Prover (Alice) computes

$E = f^z$	$E = gf^z$
$\omega, r_1, c_1 \in_R \mathcal{Z}_q$	$\omega, r_2, c_2 \in_R \mathcal{Z}_q$
$a = f^\omega$	$a = f^{r_2} E^{-c_2}$
$b = f^{r_1}(E/g)^{-c_1}$	$b = f^\omega$

(computing in mod p)

then sends a, b to the Verifier (Bob).

2. Bob responds with the challenge value $c \in_R \mathcal{Z}_q$.
3. Alice computes

$E = f^z$	$E = gf^z$
$c_2 = c - c_1$	$c_1 = c - c_2$
$r_2 = \omega + zc_2$	$r_1 = \omega + zc_1$

(computing in mod q)

then sends r_1, r_2, c_1, c_2 to the Bob.

4. Bob checks that

$$c \stackrel{?}{=} c_1 + c_2 \pmod{q}$$
$$f^{r_1} \stackrel{?}{=} b(E/g)^{c_1} \pmod{p}$$
$$f^{r_2} \stackrel{?}{=} aE^{c_2} \pmod{p}$$

Bob will accept $|x| = m+1$ if the result of running Protocol-Bit is accepted for each E_i $(i = 0, 1, \cdots, m)$. We reason about the security of the bit-commitment scheme below.

Firstly, if x and u are indeed the numbers that Alice has constructed in (1) and (2), then indeed

$$\prod_{i=0}^{m} E_i^{2^i} = g^{a_0 2^0 + a_1 2^1 + \cdots + a_m 2^m} f^{u_0 2^0 + u_1 2^1 + \cdots + u_m 2^m}$$
$$= V_1 f^u = E(x, u) \pmod{p}.$$

So if Alice is able to find a $x' \neq x \pmod{q}$ such that $E(x, u) = E(x', u')$, then it has to be $u' \neq u \pmod{q}$ and

$$\log_g f = \frac{x - x'}{u' - u} \mod q.$$

This means that Alice knows $\log_g f$, contradicting the assumption that this value is known to nobody. Therefore, x in (1) and u in (2) give the only way for Alice to demonstrate

$$V_1 f^u = E(x, u) \pmod{p}$$

3 Security Analysis

3.1 A Drawback

We remark that a cheating prover can succeed to convince a verifier that $|x| = |\hat{x}|$, where $\hat{x} \equiv x \bmod q$, q is the order of the cryptographic group used for zero-knowledge proof of knowledge of a discrete logarithm. In fact, by Mao's security argument, we know the conclusion that *Alice knows* $\log_g f$*, contradicting the assumption that this value is known to nobody*, is directly derived from the precondition that *if* x *and* u *are* ***indeed*** *the numbers that Alice has constructed in (1) and (2)*. Clearly, the precondition is unreasonable because it requires that the prover is always honest.

By the description of the scheme, we know Bob cannot force Alice to construct x as follows:

$$x = a_0 2^0 + a_1 2^1 + \cdots + a_m 2^m \qquad (*)$$

for $a_i \in \{0, 1\}$and $i = 0, 1, \cdots, m$. Actually, given an arbitrary number x, Alice only needs to compute $x = \hat{x} + kq$ such that $\hat{x} \in \mathcal{Z}_q, k \in \mathcal{Z}$. She then computes $|\hat{x}|$. Thus, Alice succeeds to convince Bob that $|x| = |\hat{x}|$ even though $|x| > |\hat{x}|$ or $|x| < |\hat{x}|$.

Note that the Proof can only show the bit-length of the residue class $\hat{x}$, where $\hat{x} \in \mathcal{Z}_q$.

3.2 The Premise in the Key Lemma Cannot Be Satisfied

Based on the above observation, we show that the premise in the following key lemma cannot be satisfied. Therefore, the whole protocol fails.

Here we relate it and its proof as follows.

Lemma. *Let* $n_1 n_2 = n \bmod q$ *and* $|n| + 2 < |q|$. *If*

$$|n_1| + |n_2| \leq |n| + 1$$

then $n_1 \mid n$ *and* $n_2 \mid n$.

Proof. *Suppose to the contrary (without loss of generality)* $n \neq 0 \bmod n_1$. *Then* $n_1 n_2 = n + kq$ *for some integer* $k \neq 0$. *Noting* $0 < n < q$, *so*

$$|n_1| + |n_2| \geq |n_1 n_2| = |n + kq| \geq |q| - 1 > |n| + 1 \tag{3}$$

contradicting the condition $|n_1| + |n_2| \leq |n| + 1$.

In fact, by the foregoing subsection analysis, we cannot show

$$|n_1| + |n_2| \leq |n| + 1 \tag{4}$$

instead can only show

$$|\widehat{n_1}| + |\widehat{n_2}| \leq |n| + 1 \tag{4'}$$

where $\widehat{n_1}, \widehat{n_2} \in \mathcal{Z}_q$, $\widehat{n_1} \equiv n_1 \bmod q, \widehat{n_2} \equiv n_2 \bmod q$. That is to say, the author [12] did not seriously distinguish the difference between (4) and (4′). The cursoriness leads him to draw a false conclusion.

Obviously, (3) does not contradict (4′). Thus, the lemma is not sound. Therefore, in the step (3-2), Bob cannot be convinced that $|P| + |Q| \leq |N| + 1$.

3.3 Alice's Trick

Now we present a simple attack against Mao's protocol of verifiable encryption of integer factorization.

Attack. Alice only needs to replace the original Data Preparation with the following.

Given N, Alice generates two numbers P_1, Q_1, such that

$$\begin{aligned} &N = P_1 Q_1 \bmod q \\ &|\widehat{P_1}| + |\widehat{Q_1}| < |N| + 1 \\ &abs(|\widehat{P_1}| - |\widehat{Q_1}|) < C \end{aligned}$$

where

$$\begin{aligned} &\widehat{P_1} \in \mathcal{Z}_q^*,\ \widehat{P_1} \equiv P_1 \bmod q \\ &\widehat{Q_1} \in \mathcal{Z}_q^*,\ \widehat{Q_1} \equiv Q_1 \bmod q \end{aligned}$$

Computes $V_1 = g^{\widehat{P_1}} \bmod p, V_2 = g^{\widehat{Q_1}} \bmod p$. Encrypts $\widehat{P_1}$ in A_1, B_1, and $\widehat{Q_1}$ in A_2, B_2 as follows:

$$(A_1 = h^{K_1}, B_1 = Y^{-K_1}\widehat{P_1}) \pmod q,$$
$$(A_2 = h^{K_2}, B_2 = Y^{-K_2}\widehat{Q_1}) \pmod q$$

where $K_1, K_2 \in_R \mathcal{Z}_r$, and Y is an agreed public key.

4 Further Discussion

Naturally speaking, the verifiable encryption scheme can be treated as a variation of a commitment scheme. More precisely, it is called a range-bounded commitment. In the past decade, there are a few schemes investigating range-bounded commitments. In Eurocrypt'98, Chan et al. [4] presented an instantiation (CFT proof for short). It's corrected soon [5] because the authors did not notice that Alice can cheat Bob if the *order* of the cryptographic group is *known* by her. Based on CFT proof, Boudot [2] constructed a popular range-bounded commitment scheme in Eurocrypt'2000, which was improved by Cao and Liu in ICICS2007 [6].

According to the above security analysis of Mao's scheme, we know the reason of the failure results also from that the order of the cryptographic group is *known* by the prover. To fix the scheme, it's sure that the order of the cryptographic group should be *unknown* by the prover. That means we should introduce another RSA modulus and base the Mao's scheme on RSA setting instead of the original ElGamal setting. Thus, an increase in the cost of modified Mao's scheme definitely occurs.

Acknowledgement

We thank the anonymous referees for their helpful suggestions. This work is supported by Science and Technology Innovation Fund of Shanghai Education Bureau.

References

1. Bellare, M., Goldwasser, S.: Verifiable partial key escrow. In: Proceedings of 4th ACM Conference on Computer and Communications Security, April 1997, pp. 78–91. ACM Press, New York (1997)
2. Boudot, F.: Efficient Proofs that a Committed Number Lies in an Interval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000)
3. Boudot, F., Traore, J.: Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery. In: Varadharajan, V., Mu, Y. (eds.) ICICS 1999. LNCS, vol. 1726. Springer, Heidelberg (1999)

4. Chan, A., Frankel, Y., Tsiounis, Y.: Easy Come–Easy Go Divisible Cash. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 561–575. Springer, Heidelberg (1998)
5. Chan, A., Frankel, Y., Tsiounis, Y.: Easy Come Easy Go Divisible Cash. Updated version with corrections, GTE Tech. Rep. (1998), http://www.ccs.neu.edu/home/yiannis/
6. Cao, Z., Liu, L.: Boudot's Range-Bounded Commitment Scheme Revisited. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 230–238. Springer, Heidelberg (2007)
7. Camenisch, J., Michels, M.: Separability and Efficiency for Generic Group Signature Schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 413–430. Springer, Heidelberg (1999)
8. Camenisch, J., Michels, M.: Proving in Zero-Knowledge that a Number is the Product of Two Safe Primes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 106–121. Springer, Heidelberg (1999)
9. Fujisaki, E., Okamoto, T.: Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 16–30. Springer, Heidelberg (1997)
10. Fiat, A., Shamir, A.: How to prove yourself: Practical solution to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
11. Guillou, L., Quisquater, J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988)
12. Mao, W.: Guaranteed Correct Sharing of Integer Factorization with Off-Line Shareholders. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 60–71. Springer, Heidelberg (1998)
13. Micali, S.: Fair public key cryptosystems. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 113–138. Springer, Heidelberg (1993)
14. Okamoto, T.: Threshold key-recovery system for RSA. In: Christianson, B., Lomas, M. (eds.) Security Protocols 1997. LNCS, vol. 1361, pp. 191–200. Springer, Heidelberg (1998)
15. Rabin, M.: Digital signatures and public-key functions as intractable as factorization. MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR–212 (1979)
16. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)

Proactive Fault Management with Virtualization for Software Aging

Thandar Thein, Sung-Do Chi, and Jong Sou Park

Korea Aerospace University,
200-1, Hwajeon-Dong Dukyang-Gu, Goyang City Gyeonggi-Do 412-791, Korea
{thandar,sdchi,jspark}@kau.ac.kr

Summary. Unplanned computer system outages are more likely to be the result of software failures than of hardware failures. Moreover, software applications executing continuously for a long period of time show a degraded performance and/or an increased occurrence rate of hang/crash failures. This phenomenon has been referred to as software aging. In this paper, we have conducted a study of virtualization technology and software rejuvenation that follows a proactive fault management approach to counter act the software aging. We present a model to evaluate the effectiveness of proactive fault management approach with the use of virtualization technology in operational software systems, and express downtime and costs due to downtime during rejuvenation in terms of the parameters in that model. We perform mathematical derivation and use SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) tool to evaluate the feasibility of our model. Our results show that proposed approach can provide uninterrupted availability of the services.

1 Introduction

Software failures are now known to be a dominant source of system outages. The consequences of software failure can lead to huge economic losses or risk to human life. Several studies and much anecdotal evidence point to software aging as a common phenomenon. Software aging is gaining in significance because of the growing economic importance of software and the fact that increasingly, software is a major part of the capital of many high-tech firms. Software aging usually observed as a progressive degradation through time, which can lead to system crashes or undesirable hang ups. Not only software used on a mass scale, but also specialized software used in high-availability and safety-critical applications suffer from aging. This phenomenon is particularly troublesome in long-running applications. Software aging has also been observed in many real software systems [3, 5, 16]. For this reason, in this paper we propose a new solution to counter act the aging problem.

Software rejuvenation technique has been widely used to avoid the occurrence of unplanned failures, mainly due to the phenomena of software aging or caused by transient failures [8]. Most current fault-tolerant techniques are reactive in nature. Proactive fault management, on the other hand, takes suitable corrective action to prevent a failure before the system experiences a fault. Software

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 189–200, 2008.
springerlink.com

rejuvenation is a specific form of proactive fault management which can be performed at suitable times. Proactive rejuvenation techniques have been studied in [2, 3, 6, 13, 17] and it is widely understood that this technique of rejuvenation provides better results, resulting in higher availability and lower costs.

Traditional fault-tolerant techniques work with recovery overhead because it carries out the restoration operation after system failure. The software rejuvenation technique works with the planed restart mechanism, which can lower the system recovery overhead by a grate extent. System availability can be further enhanced by taking a proactive approach to detect and predict an impending outage of a specific server in order to initiate planned failover in a more orderly fashion. This approach not only improves the end user's perception of service provided by the system, but also gives the system administrator additional time to work around any system capacity issues that may arise.

Although the fault in the application program still remains, performing the rejuvenation occasionally or periodically prevent failures due to that fault. Any rejuvenation typically involves an overhead, but it prevents more severe crash failures from occurring. Hence, an important issue in analyzing rejuvenation policies is to determine their usefulness in terms of availability, downtime, and cost and to provide an optimal criterion to decide when and how often to recover the system from the degraded state. The side effect of software rejuvenation is the temporary outage of service during rejuvenation. To solve this problem, we adopt the virtualization technology to provide uninterrupted service.

Virtual machine concept was first developed by IBM in the 1960's and popular in the 1970s [4]. At that time, computer systems were large and expensive, so IBM invented the concept of virtual machines as a way of time-sharing for mainframes, partitioning machine resources among different users. Virtualization is a proven software technology that is rapidly transforming the IT landscape and fundamentally changing the way that people compute. Virtualization technologies find important applications over a wide range of areas such as server consolidation, secure computing platforms, supporting multiple operating systems, kernel debugging and development, system migration, etc, resulting in widespread usage. Virtualization is a hot topic in the technology world. The technology enables a single computer to run multiple operating systems simultaneously. Many fields, such as autonomic computing, service consolidation, security and education publish results that praise the benefits of virtualization. Virtualization has proved as a successful tool for management of complex IT-environments and it is emerging as a technique to increase system reliability and availability [9, 12].

An approach for software rejuvenation based on automated self-healing techniques presented in [12], which exploit the usage of virtualization to optimize the self recovery actions. Software aging in virtual machine monitors (VMMs) has been studied in [10]. The main contribution of that paper is the development of a methodology for proactive management of software systems which are prone to aging, and specifically to resource exhaustion.

The idea proposed in this paper is to hold the multiple virtual machines which are running aging applications, and trigger the rejuvenation action of

each virtual machine when something anomalous is detected. Software rejuvenation using virtualization provides a way to remove faults and vulnerabilities at run-time without affecting system availability. By coupling proactive software rejuvenation and virtualization, significant increases in system availability and performance can be achieved.

The structure of the paper is as follows: Section 1 discusses problem issue and describes the methods to counteract the software aging problem. Section 2 addresses the related background. Section 3 presents our proposed approach. In section 4, we present a state transition model to describe the behaviors of virtualized cluster system and in the following section, the models are analyzed and experimental results are given to validate the model solution. Finally, we conclude with a summary of our results in section 5.

2 Background

2.1 Software Aging

The term software aging describes the phenomena of progressive degradation of the running software that may lead to system crashes or undesired hang ups [8]. Gradual performance degradation may also accompany software aging. Failures of both crash/hang type as well as those resulting in data inconsistency have been reported. Memory bloating and leaking, unreleased file-locks, data corruption, storage space fragmentation and accumulation of round-off errors are some typical causes of slow degradation. This undesired behavior is especially visible in long-running software such as web and application servers and enterprise always-on applications.

2.2 Software Rejuvenation

Software rejuvenation is a proactive fault management technique aimed at cleaning up the system internal state to prevent the occurrence of more severe crash failures in the future. It involves occasionally terminating an application or a system, cleaning its internal state and restarting it [14]. Some examples of cleaning the internal state of software are garbage collection, flushing operating system kernel tables and reinitializing internal data structures. Extreme example of rejuvenation might be a simple hardware reboot. Software rejuvenation is a cost effective technique for dealing with software faults that include protection not only against hard failures, but also against performance degradation as well.

2.3 Virtualization

Virtualization is a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or aggregation, partial or complete machine simulation, emulation, time-sharing, and others. A virtualization layer, thus, provides

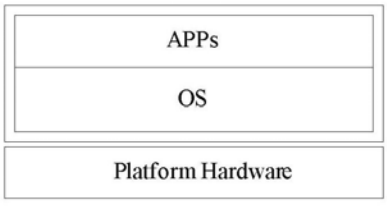


Fig. 1. (a) Without Virtualization

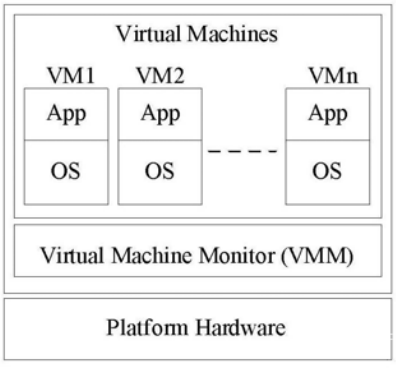


Fig. 1. (b) With Virtualization

infrastructural support using the lower-level resources to create multiple Virtual Machines (VMs) that are independent of and isolated from each other. Sometimes, such a virtualization layer is also called Virtual Machine Monitor (VMM).

The VMM is the essential part of the virtual machine implementation, because it performs the translation between the bare hardware and virtualized underlying platform: providing virtual processors, memory, and virtualized I/O devices [1]. Since all the virtual machines share the same bare hardware, the VMM should also provide appropriate protection so that each VM is an isolated replica. Traditionally, the VMM sits between the bare system hardware and operating systems. Non-virtualized and virtualized system architectures are shown in figure 1(a) and 1(b).

3 Proposed Approach

In this section we describe our proposal to offer the high availability mechanism for aging applications. This approach has been designed to use over any server or service. Our approach makes use of several concepts: a virtualization layer that is installed in every application server; the use of primary-backup scheme for application server replication; and the adoption of software rejuvenation for software aging problem. By merging all of these concepts and techniques, we can get the cost-effective and high availability solution for aging applications without requiring additional servers and load-balancing machines.

3.1 Virtualized Cluster Architecture

Figure 2 represents the conceptual architecture of our approach. It is just necessary to install a virtualization layer and install some software modules. We have adopted virtualized clustering architecture in our proposed approach. Our approach requires the creation of three virtual machines on top of the virtualization layer. In VM1, we will install software load-balancer module (VM-LB)

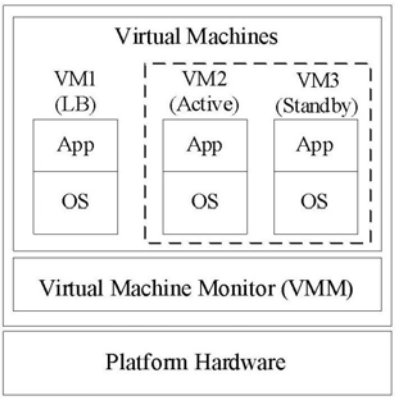


Fig. 2. VMM and VMs relationship

to collect system data from the application and some other software modules that will be responsible for the detection of software aging or other potential anomalies and to do as a coordinator of the self-recover actions. VM2 will be used to run main application server and VM3 where we create a standby replica of the application server.

We have adopted virtualized clustering architecture in our proposed approach. This setup builds an HA cluster between 3 virtual machines on a single physical machine. The cluster manager itself is running directly in the virtual machines.

3.2 Rejuvenation Process

The rejuvenation process of our approach is shown in figure 3. When software aging or some potential anomaly is detected in active VM, we should apply

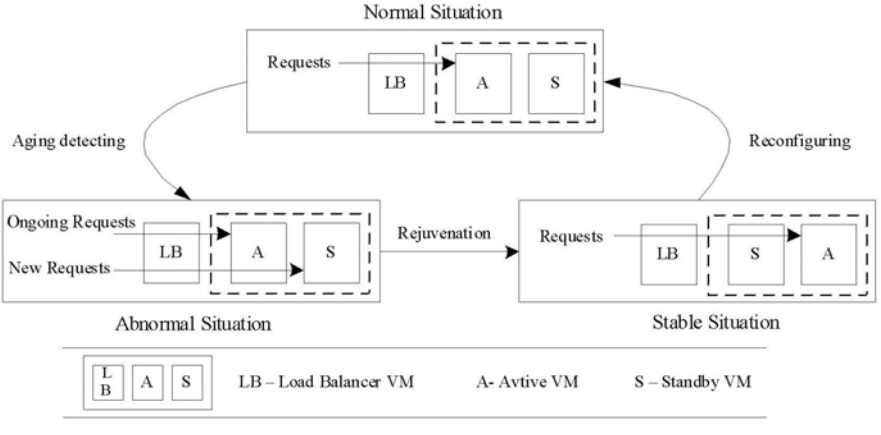


Fig. 3. Rejuvenation process

rejuvenation action. First we start the standby VM and all the new requests and sessions are migrated from the active VM to standby VM. When all the ongoing requests are finished in primary VM, then the primary VM will be rejuvenated and the VM becomes as good as new.

4 System Model and Analysis

We construct the state transition model to describe the behavior of our proposed system, and it is shown in figure 4. Suppose that the system is started with the normal operation state. The system can smoothly degrade in time. We denote failure rate (λ) and repair rate (μ) for all respective states such as error detection rate (λ_d), rejuvenation triggering rate (λ_r), switchover rate (λ_s), failure rate (λ), rejuvenation service rate (μ_r) and repair rate (μ).

The state transition diagram consists of seven states. The explanation of the states for the state transition diagram of figure 4 is as follows:

(M,1,1): VM-LB is in Monitoring state, Active VM and Standby VM are in nor-mal state
(M,D,1): VM-LB is in Monitoring state, Active VM is in detection state and standby VM is in normal state
(M,R,1): VM-LB is in Monitoring state, Active VM is in rejuvenation state and another VM is in normal state
(M,0,1): VM-LB is in Monitoring state, one VM is in failure state and one active VM is in normal state
(M,0,D): VM-LB is in Monitoring state, one VM is in failure state and active VM is in detection state
(M,0,R): VM-LB is in Monitoring state, one VM is in failure state and active VM is in rejuvenation state
(M,0,0): VM-LB is in Monitoring state, Active VM and Standby VM are in failure state.

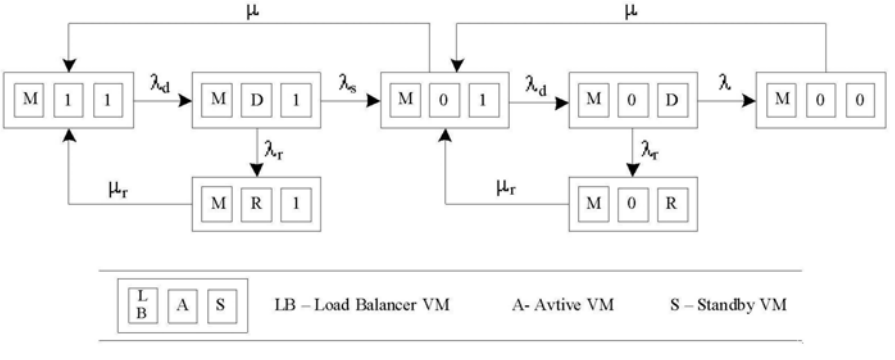


Fig. 4. A state transition diagram of system's behavior

And let the steady-state probabilities of the state of the system are as follows:

P_1=the probability that the system is in (M,1,1) state
P_2=the probability that the system is in (M,D,1) state
P_3=the probability that the system is in (M,0,1) state
P_4=the probability that the system is in (M,R,1) state
P_5=the probability that the system is in (M,0,D) state
P_6=the probability that the system is in (M,0,R) state
P_7=the probability that the system is in (M,0,0) state

The assumptions used in the modeling are as follows:

- Failure rate and repair rate of the VM are identical at all states.
- Detection rate is identical at all states.
- Rejuvenation rate is identical at all states.

After long mission time, the normal states of active VM may change to detection state with rate λ_d. In detection states, server performance is degraded and software-aging effects render the system unreliable. If a server is in detection state (M,D,1)/(M,0,D), the state can change to either a rejuvenation state (M,R,1)/(M,0,R) with rate λ_r or switchover to (M,0,1) state with rate λ_s. In failure state (M,0,0), all servers stop running and no available server remains. When software aging or some potential anomaly is detected in active VM, the system can still operate by using secondary VM and simultaneously rejuvenating the primary VM. This situation is depicted in the model of figure 4 by the transition from state (M,D,1) to (M,0,1) at rate λ_s.

Our state transition diagram can be described as a Markov process class. So we can perform steady-state analysis of the diagram easily. The steady-state balance equations of the system are as follows:

$$\mu P_3 + \mu_r P_4 = \lambda_d P_1 \tag{1}$$
$$\lambda_d P_1 = (\lambda_s + \lambda_r) P_2 \tag{2}$$
$$\lambda_r P_2 = \mu_r P_4 \tag{3}$$
$$\lambda_s P_2 + \mu_r P_6 + \mu P_7 = (\mu + \lambda_d) P_3 \tag{4}$$
$$\lambda_d P_3 = (\lambda + \lambda_r) P_5 \tag{5}$$
$$\lambda_r P_5 = \mu_r P_6 \tag{6}$$
$$\lambda P_5 = \mu P_7 \tag{7}$$

Solving the steady-state balance equations, we find,

$$P_2 = \frac{\lambda_d}{(\lambda_s + \lambda_r)} P_1 \tag{8}$$
$$P_3 = (\frac{\lambda_d}{\mu} - \frac{\lambda_r \lambda_d}{\mu(\lambda_s + \lambda_r)}) P_1 \tag{9}$$
$$P_4 = \frac{\lambda_r \lambda_d}{\mu_r(\lambda_s + \lambda_r)} P_1 \tag{10}$$

$$P_5 = \frac{\lambda_d}{(\lambda + \lambda_r)}(\frac{\lambda_d}{\mu} - \frac{\lambda_r \lambda_d}{\mu(\lambda_s + \lambda_r)})P_1 \tag{11}$$

$$P_6 = \frac{\lambda_r}{\mu_r}\frac{\lambda_d}{(\lambda + \lambda_r)}(\frac{\lambda_d}{\mu} - \frac{\lambda_r \lambda_d}{\mu(\lambda_s + \lambda_r)})P_1 \tag{12}$$

$$P_7 = \frac{\lambda}{\mu}\frac{\lambda_d}{(\lambda + \lambda_r)}(\frac{\lambda_d}{\mu} - \frac{\lambda_r \lambda_d}{\mu(\lambda_s + \lambda_r)})P_1 \tag{13}$$

The conservation equation of figure 4 is obtained by summing the probabilities of all states in the system and the sum of the equation is 1.

$$\sum_{i=1}^{n} P_i = 1 \tag{14}$$

$$P_1 = [1 + \frac{\lambda_d}{(\lambda_s + \lambda_r)}(1 + \frac{\lambda_r}{\mu_r}) + (\frac{\lambda_d}{\mu} - \frac{\lambda_r \lambda_d}{\mu(\lambda_s + \lambda_r)})(1 + \frac{\lambda_d}{(\lambda + \lambda_r)}(1 + \frac{\lambda_r}{\mu_r} + \frac{\lambda}{\mu}))]^{-1} \tag{15}$$

Using system-operating parameters, first we obtain the probability P_1, and then we calculate the probabilities of being in (M,D,1), (M,R,1), (M,0,1), (M,0,D), (M,0,R) and (M,0,0) states. The system is not available in rejuvenation state (M,0,R) and failure state (M,0,0). The availability of the system can be defined as follows:

$$Availability = 1 - (P_6 + P_7) \tag{16}$$

The expected total downtime of the system in an interval of T time units is:

$$DownTime = (P_6 + P_7) \times T \tag{17}$$

When the system is down, no service is provided and no revenue is received. There is a business cost due to service being unavailable during downtime. Predictable shutdown cost is far less than that of unexpected shutdown ($C_f \gg C_r$). If C_f is the average per unit cost of unscheduled downtime and C_r is the average per unit cost of downtime during rejuvenation, then the total expected downtime cost in an interval of T time units is:

$$DownTimeCost = ((P_6 \times C_r) + (P_7 \times C_f)) \times T \tag{18}$$

4.1 Experiments

To acquire system dependability measures like availability, downtime and downtime cost, we perform experiments using the system-operating parameters shown in table 1 [11, 14]. We perform mathematical derivation and use SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) tool to evaluate the feasibility of our model. SHARPE is a well known package [7, 15] in the field of reliability and performability. It is possible to use different kinds of models hierarchically for different physical or abstract levels of the system and to use different kinds of models to validate each other's results.

Table 1. System-operating parameters

Parameters	Values
T	1 year
λ_d	1 time/month
λ_r	1 time/month
$\frac{1}{\lambda_s}$	3 min
λ	1 time/year
μ	2 times/day
$\frac{1}{\mu_r}$	10 min
C_r	20 units
C_f	1000 units

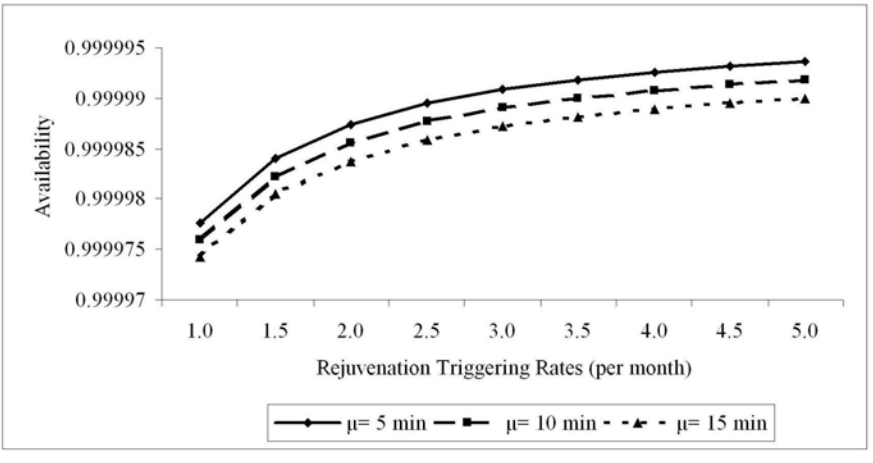


Fig. 5. Availability vs rejuvenation triggering rates and rejuvenation service rates

To examine the influence of rejuvenation triggering rates and recovery rates (rejuvenation service rates) on system availability, we set the value of recovery rate is range from 1 to 5 and rejuvenation service time is range from 5 min to 15 min. The plot of system availability in the case of performing rejuvenation is shown in figure 5.

The graph (figure 5) shows that the larger the rejuvenation triggering rates, the larger the system availability that is expected. We also find out that the faster the rejuvenation service time, the larger the availability of the system. According to the result, system availability can increase by using software rejuvenation and vir-tualized clustering technology.

The change in the downtime of virtualized clustering system with the different rejuvenation triggering rates and different rejuvenation service rates is plotted in figure 6. The change in the downtime cost of virtualized clustering system

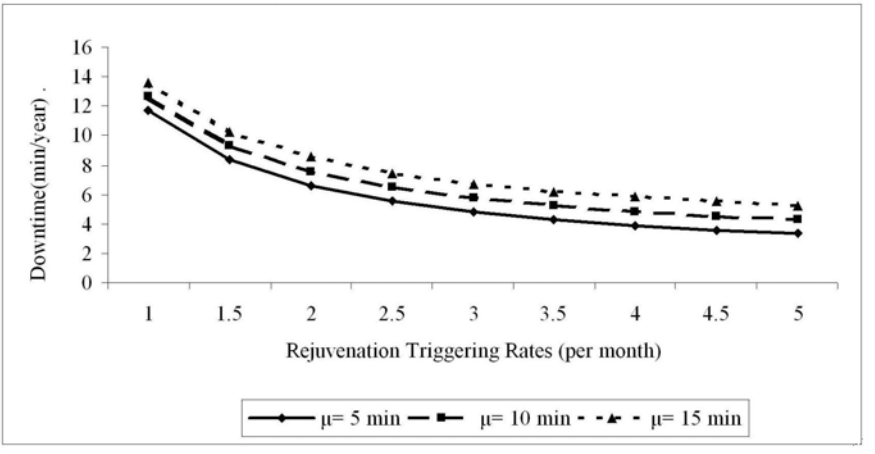


Fig. 6. Downtime vs rejuvenation triggering rates and rejuvenation service rates

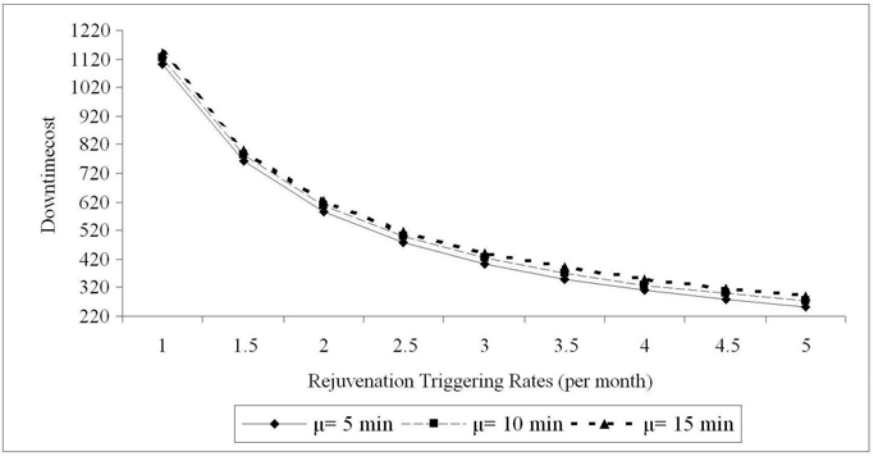


Fig. 7. Downtime cost vs rejuvenation triggering rates and rejuvenation service rates

with the dif-ferent rejuvenation triggering rates and different rejuvenation service rates is plot-ted in figure 7.

The downtime cost of scheduled shutdown is much lower than that of an unscheduled shutdown. According to the figure 6 and 7, rejuvenation can decrease both the downtime and the cost due to down time in this example and hence it is beneficial.

From our results, it is apparent that our proposed approach is a cost-effective way to build high availability system and virtualized clustering technology can improve the software rejuvenation process. For our proposed approach's validity

we use SHARPE tool. And the evaluation results through SHARPE are same with our mathematical results.

5 Conclusion

In this paper, we proposed a new approach to solve software aging problem through the use of software rejuvenation and virtualization. We present a Markov model for analyzing software rejuvenation in such continuously running applications and express availability, downtime and costs in terms of the parameters in the model. We validated our experiment results with the evaluation results through SHARPE tool. It is found that our analytical results and SHARPE results are same. Our results show that our approach can be used to prolong the availability of the services. Our approach can be applied to single-server or cluster configurations without any additional cost. Future work will include experiments with an implementation under real world conditions to verify the practical efficacy of the approach.

Acknowledgement. His research was supported by the Advanced Broadcasting Media Technology Research Center (ABRC) in Korea Aerospace University, Korea, under the Gyeonggi Regional Research Center (GRRC) support program supervised by Gyeonggi Province.

References

1. Alonso, J., Silva, L., Andrzejak, A., Silva, P., Torres, J.: High-available grid services through the use of virtualized clustering. In: Proc. of the 8th IEEE/ACM International Conference on Grid Computing, pp. 34–41 (2007)
2. Cassidy, K., Gross, K., Malekpour, A.: Advanced pattern recognition for detection of complex software aging phenomenain online transaction processing servers. In: Proc. of the Int. Conf. on Dependable Systems and Networks, pp. 478–482 (2002)
3. Castelli, V., Harper, R.E., Heidelberger, P., Hunter, S.W., Trivedi, K.S., Vaidyanathan, K., Zeggert, W.P.: Proactive management of software aging. IBM Journal of Research and Development 45(2), 311–332 (2001)
4. Creasy, R.J.: The origin of the VM/370 time-sharing system. IBM Journal of Research and Development 25(5), 483 (1981)
5. Dohi, T., Popstojanova, K.G., Vaidyanathan, K., Trivedi, K.S., Osaki, S.: Software rejuvenation modeling and applications, Springer Reliability Engineering Handbook, pp. 245–263. Springer, Heidelberg (2003)
6. Garg, S., van Moorsel, A., Vaidyanathan, K., Trivedi, K.: A methodology for detection and estimation of software aging. In: Proc. of the 9th Int. Symp. on Software Reliability Engineering, pp. 282–292 (1998)
7. Hirel, C., Sahner, R.A., Zang, X., Trivedi, K.S.: Reliability and performability modeling using SHARPE 2000. In: Proc. of the Int. Conf. on Computer Performance Evaluation: Modelling Techniques and Tools, pp. 345–349 (2000)
8. Huang, Y., Kintala, C., Kolettis, N., Fulton, N.D.: Software rejuvenation: analysis, module and application. In: Proc. Of the Fault Tolerance Computing Symp., pp. 381–390 (1995)

9. Jagarajan, A., Mueller, F., Engelmann, C., Scott, S.: Proactive fault tolerance for HPC with Xen virtualization. In: Proc. of the Int. Conf. on Supercomputing 2007, pp. 23–32 (2007)
10. Kourai, K., Chiba, S.: A fast rejuvenation technique for server consolidation with virtual machines. In: Proc. of the 37th Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN 2007), pp. 245–255 (2007)
11. Park, K., Kim, S.: Availability analysis and improvement of active/standby cluster systems using software rejuvenation. The Journal of Systems and Software 61, 121–128 (2002)
12. Silva, L.M., Alonso, J., Silva, P., Torres, J., Andrzejak, A.: Using virtualization to improve software rejuvenation. In: Proc. of the 6th IEEE Int. Symp. on Network Computing and Applications, pp. 33–44 (2007)
13. Silva, L., Madeira, H., Silva, J.G.: Software aging and rejuvenation in a SOAP-based server. In: Proc. of the IEEE Network Computing and Applications, pp. 56–65 (2006)
14. Software Rejuvenation. Department of Electrical and Computer Engineering, Duke University, `http://www.software-rejuvenation.com/`
15. Trivedi, K.S.: Symbolic hierarchical automated reliability and performance evaluator (SHARPE). In: Proc. of the Dependable System and Networks, p. 544 (2002)
16. Trivedi, K.S., Vaidyanathan, K., Postojanova, K.G.: Modeling and analysis of software aging and rejuvenation. In: Proc. of the 33rd Annual Simulation Symp., pp. 270–279 (2000)
17. Vaidyanathan, K., Trivedi, K.: A comprehensive model for software rejuvenation. IEEE Trans. on Dependable and Secure Computing 2(2), 124–137 (2005)

A Preemption Control Technique for System Energy Minimization of Weakly Hard Real-Time Systems

Smriti Agrawal, Rama Shankar Yadav, and Ranvijay

Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology, Allahabad, India
agrawal.smriti@gmail.com, rsy@mnnit.ac.in, ranvijay.mnnit@gmail.com

Abstract. This paper aims to present a general scheduling algorithm which offers lesser energy consumption for weakly hard real time systems modeled with (m, k) constraint. The weakly hard real time system consists of a DVS processor (frequency dependent) and peripheral devices (frequency independent). The energy minimization is done in two phases. In the first phase we suggest new static partitioning strategy that partitions the jobs into mandatory and optional followed by a greedy based speed assignment at the task level. Theorem is being derived to show the feasibility condition of weakly hard real–time system with modified partitioning strategy. The second phase proposes a preemption control technique that can effectively reduce the preemption impact by delaying the higher priority jobs. The simulation results and examples demonstrate that our approach can effectively reduce the overall system energy consumption (especially for systems with higher utilizations) while guaranteeing the (m, k) requirement at the same time.

Keywords: Dynamic power down, Dynamic voltage scaling, (m, k) model, Portable devices, Scheduling.

1 Introduction

Applications like multimedia such as video conferencing is being referred to as weakly hard real time where missing of some tasks to complete by deadlines degrade the quality of result however result is acceptable. For example, in real-time transmission of digitized full motion video; a source (e.g., a video camera) generates a stream of video frames at a rate of say 30 frame/sec from which at least 24 frames/sec are needed to visualize the movement of the image [17]. When transmitting such frames if sufficient processing power and network bandwidth are available then a high quality video (receiving 30 frames /sec at destination) can be projected whereas degraded quality images is received in case one or more of these frames are unable to reach within deadline. For weakly hard real time systems the assurance of minimum acceptable quality result is attained by imprecise concept [17, 18] or by (m, k) model [11]. In imprecise concept a frame has to be received at destination (may be full or portion of it) while a partial frame received is considered as dropped frame in (m, k). That is, all frames are required to be received for imprecise computation whereas certain frames may be dropped to maintain the minimum quality in (m, k) constraints. In order to ensure a deterministic quality of service (QoS) to such systems, Hamdaoui and Ramanathan [1] used the (m, k) model in which, out of k consecutive task

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 201–215, 2008.
springerlink.com

instances any m instances must meet their respective deadlines. The (m,k) model scatters the effect of m deadline misses over a window of k which is different from accepting low miss rate in which a series of frames may be lost in a burst leading to intolerant behavior in terms of missing a portion. Besides guaranteeing for QoS in terms of (m,k) designer of real time system has to take care of minimization of energy especially for portable devices.

Energy-aware computing has been realized as one of the key area for research in real time systems [20]. Energy-driven voltage scheduling algorithms have been developed to reduce system's energy consumption while satisfying the timing constraints [2, 3, 4, 5, 6, 19, 20, 21, 22, 23, 24, 25] are applicable for system having frequency dependent component (speed of the system varies with variation in its operating frequency) as resource. They will be able to reduce energy for system having frequency dependent components only. Besides frequency dependent component many systems have frequency independent components such as memory where energy-driven voltage scheduling algorithms are inadequate.

For the systems having frequency dependent component energy decreases with reduction in operating frequency and vice-versa. The energy consumption may increase while frequency is being reduced for the system having both frequency dependent as well as independent components. This is because on reducing the frequency, components which are frequency independent may be forced to be active for longer duration leading to more energy consumption. Authors [7, 8, 9, 10] reveled that the frequency dependent (processor core) consumes around 30% of total energy while frequency independent (memory and peripherals devices) account for the remaining 70% of energy consumption. Thus, the energy consumption of the frequency independent components plays a crucial role in overall energy consumption of a system. Thus, group of researcher [6, 26, 28, 32] are focused for minimization of *system energy* (energy required by frequency dependent and independent component) rather than minimization of *processor energy* only. We use the term frequency dependent component to refer a processor and frequency independent for memory or peripheral devices. The two common techniques used for minimization of system energy are *dynamic voltage scaling* (DVS) and *dynamic power down* (DPD) which will be discussed in the following subsection.

Dynamic Voltage Scaling (DVS), is based on adjusting the processor voltage and frequency on-the-fly [12, 13]. Power requirement depends on operating frequency as well as voltage i.e. the dynamic processor power is a strictly increasing convex function of the processor speed (frequency). The DVS attempts to reduce the processor speed to the extent it is possible, to obtain higher energy saving. The speed of a frequency dependent component is said to be reduced if it is either operating at lower voltage or frequency. The task response time increases with the reduced processor speed leading to the following consequences:

- a release may miss its deadline when it is feasible at higher speed.
- the longer execution time will be able to decrease the dynamic energy consumption of the processor.
- frequency independent components remain active for longer time and increase the energy consumption.
- longer execution time implies more losses in energy due to leakage current [44].

However, the task execution times do not always scale linearly with the processor speed [13, 14, 15, 16] because system may have some components (memory and i/o devices) which do not scale with the operating frequency. Thus, DVS may not be efficient (further reduction in the speed would increase the energy consumption) when the system energy is considered. To solve this problem, authors [27, 29, 30, 31] suggested a lower bound (*critical speed* which balanced the energy consumption between the processor and peripheral devices to minimize the system energy) on the processor speed to avoid the negative impact of the DVS. Niu and Quan [11] used a combined static/dynamic partitioning strategy for (m, k) model. They suggested a DVS scheduling method to reduce processor energy consumption. Beside the DVS approach authors [35, 36] suggested to switch off the system (power down) rather than to scale down the speed to reduce the energy requirement which is discussed briefly in next subsection.

Dynamic Power Down (DPD) is switching to sleep mode (low power mode) of the unused components of a system since the workload is not constant at all times. Although leaving a component (frequency dependent or independent) in idle/active state consumes power but switching to sleep mode too often may also be counter productive due to heavy context switching overheads. Thus, the DPD technique strives to balance the active and the sleeping time of the components.

Authors [32, 34, 35] used DPD to switch the processor and the peripheral devices into sleep mode based on *threshold* (minimum time for which the component may sleep for positive energy saving) value to save energy for both hard and soft real time systems. While Niu and Quan [36] proposed a DPD based scheduling method to reduce the system energy consumption for weakly hard real-time systems with (m, k) constraints.

The DPD will shut down the component based on the threshold value without taking speed into the account for all idle slots available. On the other hand, the DVS always favor to execute the task on slower speed, hence, may force a component to remain in idle/active state when it could very well sleep and save energy in DPD approach, i.e., the energy minimization by DVS is only due to variation in speed. Thus, a combination of the above two techniques is needed which could judiciously combine the positive qualities of both to provide better system energy minimization. Recently, Niu and Quan [37] have proposed such a scheduling technique for weakly hard real time systems which trades off between DVS and DPD. They have suggested a preemption control technique at the assigned speed so the application of the technique is limited.

In this paper we aim to minimize the system energy for weakly hard real time systems modeled with (m, k) constraint using a combination of DVS and DPD. The energy minimization is done in two phases, in the first phase the feasibility and energy reduction at the task level is achieved while further reduction in the energy consumption is accomplished in the improvement (second) phase at the job level. We propose a new portioning strategy to decide a job to be mandatory or optional with speed assignment for each task is done based on the greedy speed assignment technique in phase 1. While in the second phase we adopt the preemption control technique by delaying the higher priority jobs without missing its deadline. The rest of the paper is organized as follows; the next section provides a system model followed by section 3

which presents our new approach and the algorithm. The simulation results are enlisted in section 4 and the paper concludes with the section 5.

2 System Model

This paper aims to minimize the system energy consumption for a system having independent periodic task set $T = \{\tau_1, \tau_2, \tau_3 \ldots \tau_n\}$ while maintaining a minimum QoS defined by (m, k). The system consists of two types of components namely, frequency dependent (processor) and frequency independent (memory and peripheral devices). The following considerations are made:

- The frequency independent attached with the system are represented by set $A = \{a_1,\ a_2,\ a_3 \ldots\ a_N\}$ where a_i represents a memory or peripheral device. Most power management policies [37, 39] assume only one active state (to serve request) and one sleeping state (to save energy) we assume the same. Further we assume that there is no resource conflict this assumption is same as that considered in [37].
- The frequency dependent components (DVS processor) can operate at $\mathcal{N} + 1$ discrete voltage levels i.e. $V = \{v_{slp},\ v_1,\ v_2,\ v_3 \ldots v_{\mathcal{N}}\}$ where each voltage level is associated with a corresponding speed from the set $S = \{s_{slp},\ s_1,\ s_2,\ s_3 \ldots s_{\mathcal{N}}\}$ where s_1 is the lowest operating speed level measure at voltage v_1 whereas maximum speed $s_{\mathcal{N}}$ at the voltage level $v_{\mathcal{N}}$. The processor can lie in one of the three possible states namely active, idle and sleep. In the active state the processor can run at any of the speed levels between s_1 to $s_{\mathcal{N}}$, while in the idle state and sleep state it will function at speed s_1 and s_{slp} repectively.
- Each task $\tau_i \in T$ has attributes $< e_i(s_j),\ p_i,\ d_i,\ m_i,\ k_i >$ where $e_i(s_j),\ p_i$ and d_i are the computation time at the speed s_j, period and relative deadline respectively. We assume that the task relative deadline is conservative [38, 45] i.e. $d_i \leq p_i$ which is same as considered in [37]. Beside these temporal characteristics minimum QoS requirement is represented by a pair of integers $(m_i,\ k_i)$, such that out of k_i consecutive release of τ_i at least m_i releases must meet their deadline.

The symbols used in this paper are summarized in the table1 while the terms used are discussed in the next subsection.

2.1 Symbols

Table 1. Symbol Table

rel_i^j	Release time of a job τ_i^j, i.e., $rel_i^j = j * p_i$
D_i^j	Absolute deadline of a job τ_i^j, i.e., $D_i^j = j * p_i + d_i$
s_{ci}	Critical speed of the processor for the task τ_i
s_{ai}	Speed of the processor assigned to the task τ_i

Table 1. (*continued*)

a_i^j	Frequency independent component a_i is associated with task τ_j
$E_{dslp,i}^j$	Energy consumed per unit time by the device a_i associated with task τ_j in sleep state
$E_{dact,i}^j$	Energy consumed per unit time by the device a_i associated with task τ_j in active state
thd_i	DPD threshold of the device a_i
E_{pidle}	Energy consumed per unit time by the processor in the idle state
E_{pslp}	Energy consumed per unit time by the processor in the sleep state
E_{pi}	Energy consumed per unit time by the processor when running at a speed s_i ($E_{pi} = Cs_i^3$ where C is constant)
thp	DPD threshold of the processor
L	MK_hyperperiod time at which the $(m_i,\ k_i)$ pattern repeats itself, i.e., $L = LCM((k_i * p_i)\ where\ i = 1, 2\ \ldots n$ where i = 1,2,…n

2.2 Terms Used

Response time of a job $\left(\boldsymbol{R_i^j}\right)$***:*** is the response time a job τ_i^j which it needs to complete after it is released, mathematically, $R_i^{j,\gamma}(s_k) = e_i(s_k) + \sum_{h=1}^{N} H_{(h,i,j)} e_h(s_{ah})$ where $H_{(h,i,j)}$ is the number of mandatory jobs of τ_h preempting τ_i^j during the time $\left(rel_i^j, rel_i^j + R_i^{j,\gamma-1}(s_k)\right)$. The iterative equation terminates when either of the two conditions is satisfied: a) value of the two consecutive iteration is same i.e., $R_i^{j,\gamma-1}(s_k) = R_i^{j,\gamma}(s_k)$ or b) value exceeds its relative deadline i.e., $R_i^{j,\gamma}(s_k) > d_i$.

DPD threshold $(\boldsymbol{th})$***:*** In DPD policy a component is switched to a sleep state on the occurrence of idle slot to save energy. For such a switching the system has to save the state of the task at the beginning and restore the saved status at the end of sleep state (switching from sleep state to active state). These two activities incur an overhead called the DPD overhead. In case, the DPD overhead is low power down can be at each idle slot to save energy consumption whereas energy consumption increases for the case of measurable DPD overhead. In order to have a positive energy saving the component should not be switched to sleep state for duration (t) less than the power down threshold th which can be estimated as follows:

Energy consumed by a component when it remains idle during idle slot t *is* $E_{idle}t$ (1)
Energy consumed in sleep state during t

Energy consumed by the component to go into sleep state is $E_{save} * t_{save}$ and to awake is $E_{wake} * t_{wake}$ where E_{save} is the energy per unit time to save the context, E_{wake} is the energy per unit time to retrieve the context and t_{save}, t_{wake} are the time the component needs to save and wake during context switch respectively. Thus, the component can sleep for time $(t - t_{save} - t_{wake})$ and consume energy at a rate E_{slp}. Hence, the energy consumed in DPD overhead for sleep state of duration t would be

$$E_{save}t_{save} + E_{wake}t_{wake} + E_{slp}(t - t_{save} - t_{wake}) \quad (2)$$

To attain a positive energy gain the energy consumed by switching to sleep state (as measured in equation (2)) should be less than that consumed in the idle mode (as measure in equation (1)) i.e. (2)<(1)

$$\Rightarrow E_{save}t_{save} + E_{wake}t_{wake} + E_{slp}(t - t_{save} - t_{wake}) < E_{idle}t$$

In worst case when no energy gain is measured (equation (1)=(2)) then the threshold th can be estimated as

$$th = \left(E_{save}t_{save} + E_{wake}t_{wake} - E_{slp}(t_{save} + t_{wake})\right)/\left(E_{idle} - E_{slp}\right) \quad (3)$$

The threshold of each component can be estimated by equation (3).

Critical speed of the task (s_{ci})***:*** The DVS technique advocates that reduction in the speed of the frequency dependent component reduces energy consumption. This may not be true when the system is considered as a whole because lower speed leads to longer execution time for which the frequency independent components would remain active and consume energy. That is, on reduction in speed, the energy consumption first decreases then it starts increasing incase speed is further reduced. The speed at which system energy requirement is minimized for a task is called the *critical speed.* Each task in the system has its own critical speed because its computation demand and set of associated components may differ. It can be determined as follows:

Consider a task τ_i with computation time $e_i(s) = \mathbb{e}_p/s + \mathbb{e}_d$ where $\mathbb{e}_p, \mathbb{e}_d$ are the computation time frequency dependent component at the speed s and independent component respectively. Then the energy consumed by the task τ_i at speed s would be

$$E_i(s) = e_i(s) * \left(E_{pj} + \sum^{k\in A} E^i_{dact,k}\right) \quad (4)$$

In [11, 13] authors have used energy model where energy consumed by the processor is directly proportional to the cube of the operating i.e. $E_p \propto s^3$ hence, $E_p = Cs^3$ where $s \in S$.

As the task energy consumption function $E_i(s_j)$ is a strictly convex function over speed s_j it can have a single speed at which energy consumption could be minimum, this can be estimated by setting its first derivative to zero followed by the second derivative to be positive.

Thus, taking the first derivative of $E_i(s_j)$with respect to s_j as $\frac{\partial E(s)}{\partial s} = \left(-(\mathbb{e}_p/s^2)(Cs^3 + \sum^{k\in A} E^i_{dact,k})\right) + \left((\mathbb{e}_p/s + \mathbb{e}_d)(3Cs^2)\right) = 0$

$$\frac{\partial E(s)}{\partial s} = 3C\mathbb{e}_d s^4 + 2Cs^3\mathbb{e}_p - \mathbb{e}_p \sum^{k\in A} E^i_{dact,k} = 0 \quad (5)$$

By Descartes' Rule of Signs [43], there is only one positive root of the equation since the sign between two consecutive terms changes only once. This root is referred to as the critical speed of the task τ_i represented as s_{ci}.

In the following subsection we discuss the various methods for partitioning the jobs into mandatory and optional. The partitioning problem is NP-hard problems [40] hence various heuristic techniques (Red_Pattern, Even_Pattern, Rev_Pattern) can be used which are discussed below:

Deeply Red-Pattern (Red_Pattern): This pattern was proposed by Koren & Shasha [41]. Mathematically, $\pi_i^j = \begin{cases} 1, & 0 \le j \bmod k_i < m_i \\ 0, & otherwise \end{cases} \quad j = 0, 1, \ldots k_i - 1$

When π_i^j is 1, release τ_i^j is mandatory while it is optional in case 0 is assigned to π_i^j. We refer this pattern as Red_Pattern.

Evenly Distributed Pattern (Even_Pattern): Ramanathan [42] used evenly distributed pattern in which the first release is always mandatory and the distribution of mandatory and optional is evenly i.e. alternating. We refer it to as Even_Pattern. Mathematically, this can be described as

$$\pi_i^j = \begin{cases} 1, & \text{if } j = \left\lfloor \left\lceil \frac{j * m_i}{k_i} \right\rceil * \frac{k_i}{m_i} \right\rfloor \quad \text{for } j = 0, 1, \ldots . k_i - 1 \\ 0, & \text{otherwise} \end{cases}$$

Reverse evenly distribured pattern (Rev_Pattern): This pattern is a reverse of the Even_Pattern, hence the first release is always optional and the distribution of mandatory and optional is alternating. Mathematically:

$$\pi_i^j = \begin{cases} 0, & \text{if } j = \left\lfloor \left\lceil \frac{j * (k_i - m_i)}{k_i} \right\rceil * \frac{k_i}{(k_i - m_i)} \right\rfloor \; j = 0, 1, \ldots k_i - 1 \\ 1, & \text{otherwise} \end{cases}$$

This pattern was first proposed by Niu & Quan [11] and we refer it as Rev_Pattern. The following section we propose the energy minimization technique for the weakly hard real time system which was modeled in this section.

3 Energy Minimization Technique

This work is a modification of the work reported in [37] where authors have addressed the issue of system energy minimization for weakly hard real time systems modeled with the (m, k) constraint using DVS and DPD techniques. In this work they used two phase approach: in the first phase they used hybrid pattern (combination of Red_Pattern and Even_Pattern) and ensured feasibility of task set whereas preemption control is done in the second phase. The hybrid pattern allows a task to be scheduled by Red_Pattern or Even_Pattern. In both cases at least the first release of each task is mandatory (if not more e.g. (m, k)={(3, 5), (4, 7)} first two releases of both the task are mandatory with the hybrid pattern) and in phase hence, will overload the system at most, forcing it to be feasible at a higher speed requiring more energy. Therefore, to improve the performance of hybrid pattern we suggest a mixed pattern which would give a task fair chance to operate at a speed closer to its critical speed (the second release of both the task in the above example would be mandatory while the first may or may not be so. Since the second release of a task would usually be out of phase with the other releases and will not overload the system as hybrid pattern does) thus,

reduce the energy consumption (this can be observed from the example 2 given in the next subsection).

In the second phase author [37], do not allow a release to execute at a higher operating voltage than what was assigned to it in phase-1. The short coming of this approach can be seen as when a lower job is preempted by one or more higher priority jobs and the lower is incapable to fit in the slack then the finish time of the lower does not decrease with this technique and hence no reduction in energy is achieved. To over come the short coming of this technique we suggest the following two phase approach. In the first phase feasibility of the task set is ensured by the mixed pattern while further reduction in the energy consumption is endeavored in the preemption control (second) phase.

Phase 1: Task level feasibility and energy minimization

In this phase we propose a new partitioning strategy of jobs into mandatory and optional (mixed pattern) and assign speed to each task so that it is feasible. We strive to assign such speed which it is close to the critical speed of the task for which we suggest a greedy based sub-optimal speed fitting algorithm.

Both Even_Pattern and Rev_Pattern are often able to schedule the task set with lower speed than Red_Pattern leading to lesser energy consumptions. However, Red_Pattern assimilates the idle slots more effectively, hence, giving better opportunity for a component to switch to sleep state. Thus, combining the positive qualities of all the patterns we suggest the mixed pattern.

Mixed Pattern (Mix_Pattern)

Instead of assigning same pattern to all the tasks in the task set, mixed pattern (Mix_Pattern) approach will assign different type of patterns (Red_Pattern, Even_Pattern or Rev_Pattern) to each task. For example task τ_1 can be partitioned into mandatory and optional according to Red_Pattern while τ_2 and τ_3 could be assigned Red_Pattern, Even_Pattern or Rev_Pattern. Thus, yielding 3^n possible combination of pattern assignment where n is the number of the tasks in the task set. To ensure feasibility we first estimate the feasibility window (time by which the feasibility of the task set is ensured) but before that consider the example1.

Example 1: Consider a task set $T = \{< e_i(s_1),\ p_i,\ d_i,\ m_i,\ k_i >: < 20, 20, 20, 1, 3 >\}$ with speed levels $S = \{s_1, \frac{3}{2}s_1\}$. If both the task in the task set are assigned Rev_Pattern then the pattern strings would be <01011>, <001> for the task τ_1 and τ_2 respectively. If we consider the hyper period (LCM of the periods of all task) for this task set then it would be 20, but first release for both the task is optional and hence the task set is feasible. Authors [11, 37] have suggested feasibility based on the end point of the first busy interval (which is 140 in this example). It can be seen from the table 2 that the task set is feasible in first busy interval and becomes infeasible in the next busy interval (ninth release of either task is infeasible). Thus, ensuring the feasibility of the task set in first busy interval may not be sufficient to say that the task set is feasible forever. Therefore, the feasibility

window of the task set is up to the point of time (L) where entire pattern repeats (same as that considered at the start of the schedule, $t = 0$).

Table 2. Schedule for example 1

				Task τ_1		Task τ_2
j	rel_i^j	D_i^j	π_1^j	ft_1^j	π_2^j	ft_2^j
0	0	20	0	Optional	0	Optional
1	20	40	1	40	0	Optional
2	40	60	0	Optional	1	60
3	60	80	1	80	0	Optional
4	80	100	1	100	0	Optional
5	100	120	0	Optional	1	120
6	120	140	1	140	0	Optional
End of first busy interval						
7	140	160	0	Optional	0	Optional
8	160	180	1	(resource conflict)	1	(resource conflict)

The feasibility of this pattern is ensured by the following theorem:

Theorem 1: Let $\mathbb{M} = R \cup E \cup \Re$ where R, E and represents the mandatory job sets according to the Red_Pattern, Even_Pattern and Rev_Pattern respectively. Also $W^R(t_s, t_e), W^E(t_s, t_e)$ and $W^{\Re}(t_s, t_e)$ represents the workload that arrive at or after time t_s have to finished before or at t_e. Then all the mandatory jobs can meet their deadlines iff, $W^R(t_s, t_e) + W^E(t_s, t_e) + W^{\Re}(t_s, t_e) \leq t_e - t_s$ for every busy interval over $(0, L)$, i.e., $\forall (t_s, t_e) \in (0, L)$.

Proof: The workload is maximized when two mandatory releases are in phase. According to the Rev_Pattern the first release would always be optional. Thus, when it is combined with any other pattern then the workload will not be maximum in the first busy interval, therefore ensuring the feasibility in the first busy interval is not sufficient to ensure the feasibility of the task set. Hence, the feasibility of each busy interval up to MK_hyperperiod (L) (from which they repeat) is required to ensure the feasibility of the task set. ■

The mixed pattern generates possible combination of Red_Pattern, Even_Pattern and Rev_Pattern. We assume the task set is feasible if any one of these combination is feasible at the maximum speed according to theorem1. To reduce the energy consumption so that speed assigned to each task tends to the critical speed is achieved by the following speed fitting technique.

Speed Fitting

We first estimate the critical speed of each task s_{ci} and schedule the task set for MK_hyperperiod (L). If all the mandatory releases of the task set are feasible then we go to the next phase to further reduce the energy consumption i.e. improvement phase. But in case one or more releases are infeasible at critical speed then a speed is to be selected such that its energy requirement is minimum (may be more than energy required at the critical speed). The authors [11] used exhaustive search based on branch and bound concept where they considered all the possible cases. Though the scheme is offline it needs high computation demand (number of speed level raise to power to number of task) even for small problem set. In order to reduce the complexity we propose a greedy based speed fitting that utilizes the concept of choosing to increment the speed of a task for which the increment in energy requirement is least. In worst case our greedy approach requires same complexity as received in [11].

However, on average it requires lesser number of searching. In the best case it can assign the speed in a single iteration. The algorithm is stated below:

// Greedy approach based speed fitting algorithm
Algorithm speed_fitting(task set T)
Begin
1. For all the task $\tau_i \in T$
   Do
   a. Compute the critical speed for each task s_{ci}
   b. Initialize ω_i with the speed index of s_{ci}
   c. Assign $s_{ai} = s_{\omega_i}$ (which is same as s_{ci})
      Repeat
2. While (not feasible) // according to theorem1
      Do
   a. For all task $\tau_i \in T$
   Do
      i. If $(\omega_i < \mathcal{N})$
         1. Compute $\Delta_i = \left(\left(\left(E_{p\omega_i+1} e_i(s_{\omega_i+1})\right) - \left(E_{p\omega_i} e_i(s_{\omega_i})\right)\right) m_i\right) / p_i k_i$
            Else
         1. $\Delta_i = \infty$
   Repeat
   b. Select a task τ_i with smallest Δ_i
   c. If $(\omega_i < \mathcal{N})$ // maximum number of speed levels
      i. $\omega_i = \omega_i + 1$
      ii. $s_{ai} = s_{\omega_i}$
      iii. Goto step 2
         Else
      i. Goto step 2b. to select next smallest Δ_i
         Repeat
End

Table 3. Schedule for example 2

$\tau_1 = <(900, 10), 20, 20, 2, 3>$, $< a_1^1,\ 35,\ 648000,\ 0.0 >$

j	rel_1^j	D_1^j	**Hybrid Pattern**			**Mixed Pattern**		
			π_1^j	s_{a1}	ft_1^j	π_1^j	s_{a2}	ft_1^j
0	0	20	1	180	(0, 10)	1	60	(0, 20)
1	20	40	1	180	(20, 30)	1	60	(0, 40)
2	40	60	0	180	Optional	0	60	Optional

$\tau_2 = <(900, 10), 20, 20, 1, 3>$, $< a_2^2,\ 35,\ 648000,\ 0.0 >$

x	rel_2^x	D_2^j	**Hybrid Pattern**			**Mixed Pattern**		
			π_1^x	s_{a2}	ft_2^x	π_2^x	s_{a2}	ft_2^x
0	0	20	1	180	(10, 20)	0	60	Optional
1	20	40	0	180	Optional	0	60	Optional
2	40	60	0	180	Optional	1	60	(40, 60)

The speed fitting algorithm, consequently assigns the speeds to each task in such way that the task set becomes feasible. Niu & Quan [37] proposed a hybrid pattern which used the Even_Pattern and Red_Pattern simultaneously, for a task set whose performance is compared in the example 2.

Example2: Consider a task set $T = \{< (e_i = \mathbb{e}_{p,i}, \mathbb{e}_{d,i}),\ p_i,\ d_i,\ m_i,\ k_i > :< (900, 5), 20, 20, 2, 3 >, < (900, 5), 20, 20, 1, 3 >\}$ with devices $A = \{< a_i^j,\ thd_i,\ E_{dact,i}^j,\ E_{dslp,i}^j >: < a_1^1,\ 35,\ 648000,\ 0.0 >, < a_2^2,\ 35,\ 648000,\ 0.0 >\}$. The DVS processor can work at speed levels $S = \{0, 60, 90, 180\}$ and threshold $thp = 35$.

The critical speed for τ_1 and τ_2 would be $s_{c1} = s_{c2} = 60$. Thus the execution time $e_i(s_{ci}) = (900/60) + 5 = 20$ and the MK_Hyperperiod $L = LCM(20 * 3,\ 20 * 3) = 60$. The schedule and the energy consumption by both hybrid and mixed pattern can be seen from the table 2.

The percent of reduction in energy consumption 64.9%. Thus, performance of mixed pattern is better than the hybrid pattern. In the following section we improve the schedule by preemption control.

Phase-2: Preemption control

After performing speed fitting in first phase this phase will improve the schedule by preemption control. When a job is scheduled on the processor then the associated devices are switched to active state in which they remain till it completes. Thus, if a lower priority job is preempted by the higher priority job then the associated device remain active and consume energy for the time for which the job is preempted. This extra consumption in the energy can be reduced by delaying the higher priority job if possible and completing the execution of the lower priority job in the meanwhile (laxity).

The higher priority preempting job can be delayed up to its laxity available so that it does not miss its deadline. This laxity can be estimated as follows: $laxity_h^x = D_h^x - R_h^x(s_{ah})\ \forall \tau_h^x$ which preempts τ_i^j such that $rel_h^x > rel_i^j$. Thus, a job τ_i^j can be executed without preemption or the slack generated by delaying the execution of the higher priority job would be $slack_i^j = min_{\tau_h^x \in H_{(h,i,j)}}(rel_h^x + laxity_h^x)$. Hence, the time available for execution non-preemptively by the lower job would be $Ta_i^j = min\left(\left(\left(rel_h^x - rel_i^j\right) + slack_i^j\right), d_i\right)$ where $rel_i^j < min_{\tau_h^x \in H_{(h,i,j)}}(rel_h^x)$ when no higher job preempts then $rel_h^x = \infty$. The algorithm preemption_control would fit a lower job in the available time Ta_i^j.

Algorithm preemption_control(task set T)
Begin

1. For each mandatory job τ_i^j arriving during any L
Do
- **a.** Compute Ta_i^j for each job τ_i^j
- **b.** If $\left(Ta_i^j \geq e_i(s_{ai})\right)$
  - **i.** Execute τ_i^j non-preemptively for $\left(rel_i^j, rel_i^j + e_i(s_{ai})\right)$
  - **ii.** Goto step 2
- **c.** Else
  - **i.** Initialize ω with the speed index of s_{ai}
  - **ii.** While $\left(\left(Ta_i^j \geq e_i(s_\omega)\right)\ AND\ \omega < \mathcal{N}\right)$
    Do
    1. $\omega = \omega + 1$
    Repeat
  - **iii.** Assign the speed to job τ_i^j as $s_{ai}^j = s_\omega$
  - **iv.** Execute τ_i^j non-preemptively for $\left(rel_i^j, rel_i^j + e_i(s_{ai}^j)\right)$

2. Update the laxity of each of the higher jobs and execute them at lower speed (up to the critical speed) if the laxity allows.

End

The algorithm preemption_control would decrease the energy consumption by executing the lower priority job non-preemptively. When a lower job is executed at a higher speed (than assigned to it) to finish within the available time Ta then it would create room for the higher priority jobs who were preempting it to execute at a lower speed to save energy. Thus, energy saving is achieved by finishing the lower priority job earlier (preemption control to favor DPD) and executing the higher priority jobs at lower speed (to favor DVS). However, when the proposed approach requires more energy than without it we do not favor preemption control and execute the job with preemption.

4 Simulation Results

Table 4. Simulation Parameters

Parameter	Condition	Range
UTh Utilization Threshold	Is assigned	0.01
u_i Utilization	If $U - \sum u_{i-1} \geq UTh$ the select a uniform random number	$(0, U - \sum u_{i-1}]$
	If $U - \sum u_{i-1} < UTh$ then assign	$u_i = U - \sum u_{i-1}$
e_i worst case execution time	select a uniform random number	(0,100]
p_i period	select a uniform random number	(0,1000]
d_i deadline	select a uniform random number	$[e_i, p_i]$
k_i	Is a random integer selected uniformly	[1,10]
m_i is the number of mandatory jobs in k_i	Assigned a value	$\lfloor u_i p_i k_i / e_i \rfloor$
thp processor threshold	select a uniform random number	[0, 200]

This section compares the performance of our proposed mixed pattern based preemption control (MIX) with the hybrid pattern preemption control technique (HYB) used in [37]. All simulation results are computed on a DVS processor with operating speed level set as $S = \{0,\ s_1,\ s_2, \ldots\ s_{10}\}$ where s_i is a uniform random number generated in the interval $[10, 200]$. We consider three types of devices (with threshold range generated uniformly over (0, 200)) with multiple instances forming a pool of devices. For a task devices are randomly selected from this pool. Rate of energy consumption for a device is computed based on the energy required by the processor at the maximum speed, i.e., $E_{dact,i} = \text{þ}E_{p10}$ where þ is a uniform random number in the range [0.1, 10]. The task set $T = \{\tau_1,\ \tau_2,\ \ldots \tau_n\}$ with (m, k) utilization U (i.e. $\sum m_i e_i / p_i k_i$ a uniform random number in the range (0, 1]) are generated. The other parameters are summarized in the table 3.

The key parameter, measured for simulation is energy consumed during one MK_hyperperiod. The result reported is the average of ten task sets. The following section deals with the variation in energy with component threshold, task set utilization. It can be seen from the simulation results that mixed pattern approach carefully balances the energy consumption of the frequency dependent and inpendent components. For higher utlization range (0.8-1) the reduction in energy consumption by the MIX approach is aprroximately 25%, while for lower utlizations the gain is approximately 20% (refer figure-1). As can be seen from the figure-2 at lower

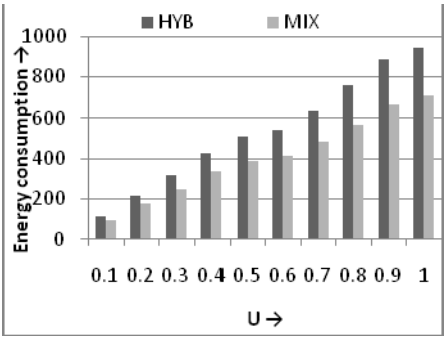


Fig. 1. Energy consumption Vs. Utilization

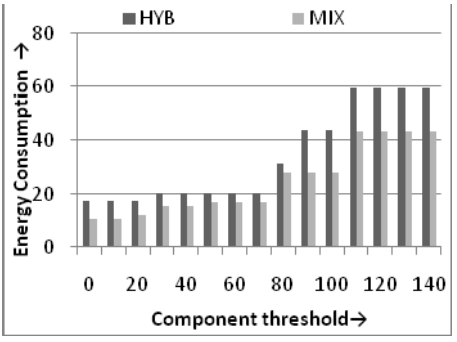


Fig. 2. Energy Vs. Component threshold

threshold values the reduction in energy by MIX approach is around 24% while it is still better for higher threshold range (80-140) of approximately 30%.The improvement achieved by MIX approach is due to the fact, that when more than one higher job preempts the lower job and there is not sufficent room to complete the lower job the HYB fails to improve the finish time of the lower so the energy consumption with it reamins estially the same as was received without it. However, the MIX approach is better equipt to finish the lower job earlier to save the energy consumed by its devices to remain active and also executes the higher priority jobs at lower speed (up to the crtical speed) to save energy.

5 Conclusion

In this paper we presented a general scheduling algorithm which minimizes the system energy consumption for weakly hard real-time system while maintaining the (m, k) guarantee. The system consists of a DVS processor (capable of operating at various frequencies) and frequency independent peripheral devices. We proposed a two phase scheduling algorithm where in the first phase we propose a mixed pattern which is a refinement of the existing partitioning strategies used by various authors [36, 37, 42]. The modified partitioning approach utilizes the greedy based speed assignment concept which is faster than the branch and bound technique suggested by authors in [11]. While in the second phase we proposed a preemption control strategy which looks at the job level to adjust the speed of the job based on the laxity to further reduce the energy consumption. Thus, the proposed algorithm is capable of performing better in the scenario when the system is overloaded (utilization is high) or the threshold of the components are high.

References

[1] Hamdaoui, M., Ramanathan, P.: A dynamic priority assignment technique for streams with (m, k)-firm deadlines. IEEE Trans. Compute. 44(12) (December 1995)

[2] Moss'e, D., Aydin, H., Childers, B., Melhem, R.: Compiler-Assisted Dynamic Power-Aware Scheduling for Real-Time Applications. In: Workshop on Compiler and OS for Low Power (2000)

[3] Qiu, Q., Wu, Q., Pedram, M.: Dynamic Power Management in a Mobile Multimedia System with Guaranteed Quality-of-Service. In: ACM/IEEE Design Automation Conf. (2001)
[4] Qu, G., Potkonjak, M.: Power Minimization Using System-Level partitioning of Applications with Quality of Service Requirements. In: IEEE Int'l Conf. on Comp-Aided Design (1999)
[5] Quan, G., Hu, X.: Energy Efficient Fixed-Priority Scheduling for Real-Time Systems on Variable Voltage Processors. In: 38th IEEE/ACM Design Automation Conference (2001)
[6] Hua, S., Qu, G.: Energy-Efficient Dual-Voltage Soft Real-Time System with (m, k)-Firm Deadline Guarantee. In: CASES 2004, Washington, DC, USA, September 22–25 (2004)
[7] Doherty, L., Warneke, B., Boser, B., Pister, K.: Energy and performance considerations for smart dust. Int'l Journal of Parallel Distributed Systems and Networks (2001)
[8] Douglis, F., Krishnan, Marsh, B.T.: The power-hungry disk. USENIX (1994)
[9] Viredaz, M.A., Wallach, D.: Power evaluation of a handheld computer. IEEE Micro., (2003)
[10] Zedlewski, J., Sobti, S., Garg, N., Zheng, F., Krishnamurthy, A., Wang, R.: Modeling hard-disk power consumption. In: FAST 2003, pp. 217–230 (2003)
[11] Niu., L., Quan, G.: Energy minimization for real time systems with (m, k)- guarantee. IEEE Trans. On Very large scale integrated (VLSI) systems 14(7) (July 2006)
[12] Weiser, M., Welch, B., Demers, A., Shenker, S.: Scheduling for Reduced CPU energy. In: USENIX Symposium on Operating Systems Design and Implementation (1994)
[13] Aydin, H., Devadas, V., Zhu, D.: System-level Energy Management for Periodic Real-Time Tasks. In: 27th IEEE International Real-Time Systems Symposium (RTSS 2006) (2006)
[14] Bini, E., Buttazzo, G.C., Lipari, G.: Speed Modulation in Energy-Aware Real-Time Systems. In: Proc. of the 17th Euromicro Conference on Real-Time Systems (ECRTS) (2005)
[15] Choi, K., Soma, R., Pedram, M.: Fine-grained dynamic voltage and frequency scaling for precise energy and performance trade-off based on the ratio of off-chip access to on-chip computation times. In: Design, Automation and Test in Europe (2004)
[16] Seth, K., Anantaraman, A., Mueller, F., Rotenberg, E.: FAST: Frequency-Aware Static Timing Analysis. In: 24th IEEE Real-Time System Symposium (2003)
[17] Huang, X., Cheng, A.M.K.: Applying Imprecise Algorithms to Real-Time Image and Video Transmission. In: Real-Time Technology and Applications Symp., Chicago (May 1995)
[18] Chen, X., Cheng, A.M.K.: An Imprecise Algorithm for Real-Time Compressed Image and Video Transmission. In: Sixth International Conference on Computer Communications and Networks, Proceedings, Las Vegas, NV, USA, pp. 390–397
[19] Yao, A.F., Demers, A., Shenker, S.: A scheduling model for reduced CPU energy. In: Proc. AFCS, pp. 374–382 (1995)
[20] Niu, L., Quan, G.: Energy-Aware Scheduling for Real-Time Systems With (m; k)-Guarantee. Dept. Comput. Sci. Eng., Univ. South Carolina, Tech. Rep. (2005)
[21] Bernat, G., Burns, A.: Combining (n;m)-hard deadlines and dual priority scheduling. In: Proc. RTSS, pp. 46–57 (December 1997)
[22] Kim, W., Kim, J., Min, S.L.: A dynamic voltage scaling algorithm for dynamic-priority hard real-time systems using slack analysis. In: Proc. DATE, p. 788 (2002)
[23] Saewong, S., Rajkumar, R.: Practical Voltage-Scaling for Fixed-Priority Real-time Systems. In: Proceedings of the IEEE Real-Time and Embedded Tech. and App. Symp. (2003)
[24] Pillai, P., Shin, K.G.: Real-time Dynamic Voltage Scaling for Low-power Embedded Operating Systems. In: Proceedings of the ACM Symposium on OS Principles (2001)

[25] Aydin, H., Melhem, R., Moss'e, D., Mejia-Alvarez, P.: Dynamic and Aggressive Power-Aware Scheduling Techniques for Real-Time Systems. In: RTSS (2001)
[26] Fan, X., Ellis, C., Lebeck, A.: The Synergy between Power aware Memory systems and Processor Voltage. In: Workshop on Power-Aware Computing Systems (December 2003)
[27] Jejurikar, R., Gupta, R.: Dynamic voltage scaling for system-wide energy minimization in real-time embedded systems. In: ISLPED (2004)
[28] Zhuo, J., Chakrabarti: System level energy efficient dynamic task scheduling, In: DAC 2005 (2005)
[29] Kim, W., Kim, J., Min, S.: Preemption aware dynamic voltage scaling in hard real time systems. In: ISLPED (2004)
[30] Kim, M., Ha, S.: Hybrid run-time power management technique for real-time embedded system with voltage scalable processor. In: OM 2001, pp. 11–19 (2001)
[31] Jejurikar, R., Gupta, R.: Dynamic voltage scaling for system-wide energy minimization in real-time embedded systems. In: ISLPED (2004)
[32] Cheng, H., Goddard, S.: Online energy-aware i/o device scheduling for hard real-time systems. In: DATE (2006)
[33] Wang, H.M., Choi, H.S., Kim, J.T.: Workload-Based Dynamic Voltage Scaling with the QoS for Streaming Video. In: 4th IEEE Int'l Symp. on Electronic Design, Test & App. (2008)
[34] Rong, P., Pedram, M.: Hierarchical power management with application to scheduling. In: ISLPED (2005)
[35] Swaminathan, V., Chakrabarty, K.: Pruning-based, energy optimal, deterministic i/o device scheduling for hard real-time systems. Trans. on Embedded Computing Sys. 4(1) (February 2005)
[36] Niu, L., Quan, G.: System-wide dynamic power management for multimedia portable devices. In: IEEE International Symposium on Multimedia (accepted, 2006)
[37] Niu, L., Quan, G.: Peripheral-Conscious Scheduling on Energy Minimization for Weakly Hard Real-time Systems. In: DATE 2007 (2007)
[38] Cucu, L., Goossens, J.: Feasibility Intervals for Multiprocessor Fixed-Priority Scheduling of Arbitrary deadline Periodic Systems. In: DATE 2007 (2007)
[39] Lu, Y.H., Micheli, G.D.: Comparing system-level power management. IEEE Design and Test of Computers (March-April 2001)
[40] Quan, G., Hu, X.: Enhanced fixed-priority scheduling with (m,k)-firm guarantee. In: RTSS, pp. 79–88 (2000)
[41] Koren, G., Shasha, D.: Skip-over: Algorithms and complexity for overloaded systems that allow skips. In: Proc. RTSS, p. 110 (1995)
[42] Ramanathan, P.: Overload management in real-time control applications using (m; k)-firm guarantee. IEEE Trans. Parallel. Distrib. Syst. 10(6), 549–559 (1999)
[43] http://www.purplemath.com/modules/drofsign.htm
[44] Jejurikar, R., Pereira, C., Gupta, R.: Leakage aware dynamic voltage scaling for real-time embedded systems. In: Proc. of the Design Automation Conf., pp. 275–280 (2004)
[45] Baruah, S., Fisher, N.: Global fixed-priority scheduling of arbitrary-deadline sporadic task systems. In: Proceedings of the 11th International Conference on Principles of Distributed Systems, Guadeloupe, French West Indies (December 2007)

Text Classification by Relearning and Ensemble Computation

Naohiro Ishii[1], Takahiro Yamada[2], and Yongguang Bao[3]

[1] Dept. of Applied Information Science, Aichi Institute of Technology
1247 Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392, Japan
ishii@aitech.ac.jp
[2] Dept. of Network Engineering, Aichi Institute of Technology
1247 Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392, Japan
v06723vv@aitech.ac.jp
[3] Aichi Information System, Sumiyoshicho 3-2, Kariya, Japan
baoyg_860@hotmail.com

Abstract. The k-nearest neighbor(k-NN) is improved by applying the distance functions with relearning and ensemble computations to classify text data with the higher accuracy values. The proposed relearning and combining ensemble computations are an effective technique for improving accuracy. We develop a new approach to combine kNN classifier based on weighted distance function with relearning and ensemble computations. The combining algorithm shows higher generalization accuracy, compared to other conventional algorithms. First, to improve classification accuracy, a relearning method with genetic algorithm is developed. Second, ensemble computations are followed by the relearning. Experiments have been conducted on some benchmark datasets from the UCI Machine Learning Repository.

Keywords: Relearning, Text Classification, kNN, Ensemble Computation.

1 Introduction

Broad band networks have made a great progress for much data communication and much information transactions in the computer networks. Though the data is becoming greatly large in the volume, the machine classification of text data, is not easy under these computing circumstances. The kNN[1, 4] is a simple and effective method among instance-based learning algorithms. However, the kNN algorithm has several shortcomings. It is expected to provide good generalization accuracy by the kNN methods for a variety of real-world classification tasks as text classification and applications. Then, improving accuracy and performance of classifiers by the kNN, is still attractive to many researchers. In this paper, first, we present a new approach of relearning to improve the kNN classifiers based on distance functions with weights, which improve the performance of the k-nearest neighbor classifier. The weights of attributes of data, are optimized by applying genetic algorithm. Second, combining ensemble computation is developed as a final classifier, which is followed by relearning mistaken classified data in the process. To improve the accuracy, not only the training data but also the testing data are important. The proposed method is developed from the 10-fold cross-validation method. Thus, it is shown that the relearning and combining procedures developed here,

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 217–226, 2008.
springerlink.com

are effective to improve the classification accuracy based on the kNN. The proposed re-learning and ensemble computation method shows a higher generalization accuracy, compared to other conventional learning algorithm.

2 Classification by Different Distance Functions

2.1 Distance Functions

The choice of distance function influences the bias of the k-nearest neighbor(kNN) classification. The most commonly used functions is the Euclidean Distance function (Euclid), which is defined as:

$$D(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

,where x and y are two input vectors (one typically being from a stored instance, and the other an input vector to be classified) and m is the number of input variables (attributes) in the application.

One way to handle applications with both continuous and nominal attributes is to use a heterogeneous distance function that uses different attribute distance functions on different kinds of attributes. The Heterogeneous Euclidean-Overlap Metric (HEOM) uses the overlap metric for nominal attributes and normalized Euclidean distance for linear attributes. This function defines the distance between two values x and y of a given attribute a as:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a,y_a)^2},$$

where

$$d_a(x,y) = \begin{cases} 1, & if\ x\ or\ y\ is\ unknown,\ else \\ overlap(x,y), & if\ a\ is\ nominal,\ else \\ \frac{|x-y|}{man_x - main_a}, & \end{cases}$$

and function overlap is defined as:

$$overlap(x,y) = \begin{cases} 0, & x = y \\ 1, & other \end{cases}$$

The Value Difference Metric (VDM), introduced by Stanfill and Waltz (1986)[7], is an appropriate distance function for nominal attributes. A simplified version of the VDM (without the weighting schemes) defines the distance between two values x and y of an attribute a as:

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q = \sum_{c=1}^{C} \left| P_{a,x,c} - P_{a,y,c} \right|^q,$$

where $N_{a,x}$ is the number of instances in the training set T that have value x for attribute a; $N_{a,x,c}$ is the number of instances in T that have value x for attribute a and output class c; C is the number of output classes in the problem domain; q is a constant, usually 1 or 2; and $P_{a,x,c}$ is the conditional probability that the output class is c given that attribute a has the value x, i.e., $P(c|xa)$. The $P_{a,x,c}$ is defined as:

$$P_{a,x,c} = \frac{N_{a,x,c}}{N_{a,x}},$$

where $N_{a,x}$ is the sum of $N_{a,x,c}$ over all classes, i.e.,

$$N_{a,x} = \sum_{c=1}^{C} N_{a,x,c}$$

and the sum of $P_{a,x,c}$ over all C classes is 1 for a fixed value of a and x.

In [10], Wilson and Martinez proposed three new alternatives to overcome the weakness of VDM. The one is a Heterogeneous Value Difference Metric(HVDM) that uses Euclidean distance for linear attributes and VDM for nominal attributes. The other two distance functions are the Interpolated Value Difference Metric (IVDM). Wilson and Martinez also proposed a generic version of the VDM distance function, called the discretized value difference metric (DVDM).

2.2 Distance Functions for Optimization

The distance function in HEOM is defined in the previous section 2.1. To characterize the respective distance function for the training data, the weighted distance function is proposed in this paper as follows,

$$WeightedHEOM(x,y) = \sqrt{\sum_{i=1}^{n} \omega_i \times (x_i - y_i)^2},$$

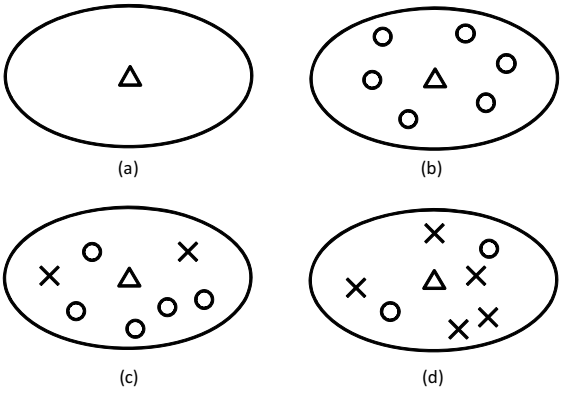


Fig. 1. Typical tolerant rough set

where shows the weight of the i-th component of the data and . Here, the weights are normalized as follows,

$$\sum_{i=1}^{n} \omega_i = 1, \quad \omega_i \geq 0$$

The problem, here is how to derive the optimized weights $\{\omega_i\}$. The optimized weights of the distance function, are computed by applying Genetic Algorithm that uses selection, recombination and mutation operations based on natural selection and biological genetics.

The text data is classified into two set of data in this study. One is the training data, which is assigned the given class in advance. The other is the testing data,
which is determined by the weighted distance and the kNN computations. In this paper, the training data is made on the tolerant rough set, which is shown in Fig. 1. The tolerant rough set is presented in our previous studies[2, 3, 11].

3 Relearning and Ensemble Computation

To improve the classification accuracy, the similar data in the same class, will be important. The mistaken classified data will have a cue to improve the accuracy. Then, the mistaken classified data, is applied again in the relearning process, which is proposed in the following section.

3.1 Relearning Computation

A relearning computation is shown in the schematic diagram in Fig.2. By using the leaning data, the first learning process is carried out by the learnt data. The classification process is done by using the testing data in (a) in Fig.2. The classified testing data in (a) is compared in the correctly classified table (b), given in advance. The misclassified data, instance 1 and instance 2, are applied in the second learning process as shown in Fig.2.

3.2 Experimental Results for Relearning Computation

The proposed relearning method was applied to the UCI depository data for the experimentations by using different distance functions, HEOM, HVDM, DVDM and IVDM described in the section 2 . To compare the relearning method with our previous method without the relearning process, the experimental results are shown in Table 1. In Table 1, the column Aver. shows the accuracy results byour previous method without relearning process and the column Re-learning shows the computational results by the proposed method with the relearning process.

In Fig.3, the experimental results of the classification accuracy without and with relearning process, are shown for the comparison. The relearning process increases the classification accuracy of the data, largely. The distance function IVDM shows higher accuracy value among distance functions.

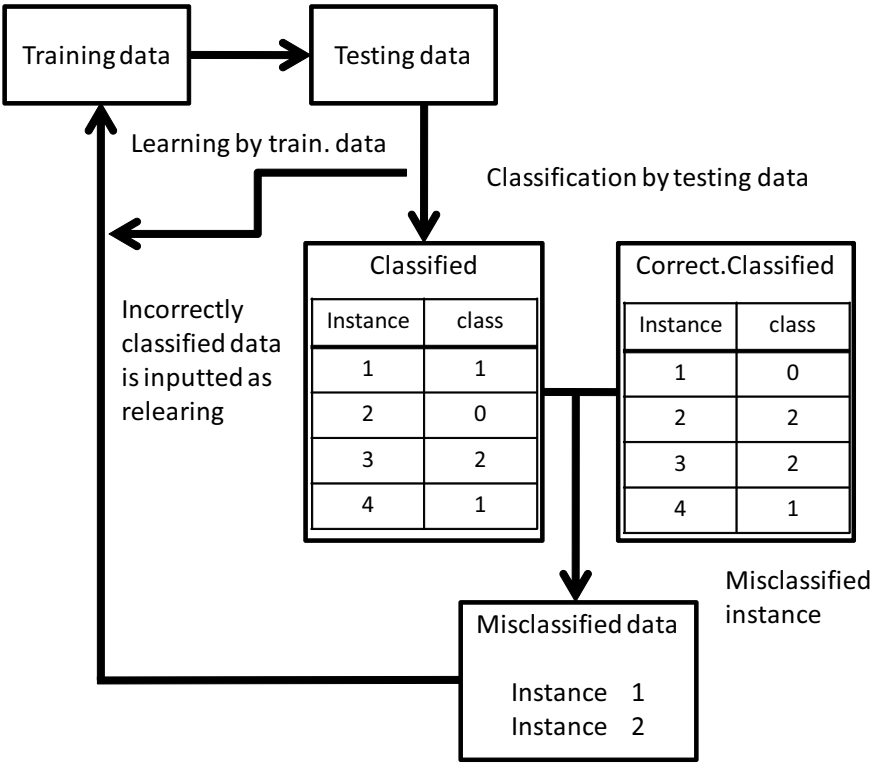


Fig. 2. Relearning process

Table 1. Experimental accuracy results by method without and with relearning process

DataSet	HEOM		HVDM		DVDM		IVDM	
	Aver.	Re-learning	Aver.	Re-learning	Aver.	Re-learning	Aver.	Re-learning
Breast	95.86	**97.51**	95.11	**96.84**	96.37	**97.87**	96.39	**97.42**
Bridges	58.39	**72.07**	60.49	**72.16**	60.09	**81.14**	58.43	**77.54**
Flag	56.32	**70.27**	58.83	**82.11**	55.30	**80.37**	54.09	**81.03**
Glass	75.33	**79.57**	71.76	**85.86**	58.66	**66.19**	77.27	**89.95**
Heart	80.04	**87.63**	80.35	**87.95**	82.36	**88.52**	80.20	**87.11**
Hearttlb	86.56	**92.00**	72.70	**84.71**	85.55	**90.05**	85.90	**90.59**
Heartswi	95.40	**97.77**	96.81	**97.00**	96.18	**97.85**	95.66	**98.49**
Hepatit	81.83	**90.05**	79.12	**89.43**	80.66	**91.35**	86.69	**90.66**
Promot	80.36	**90.54**	88.11	**97.61**	88.97	**97.63**	88.69	**95.52**
Average	**78.90**	**86.38**	**78.14**	**88.19**	**78.24**	**87.89**	**80.37**	**89.81**

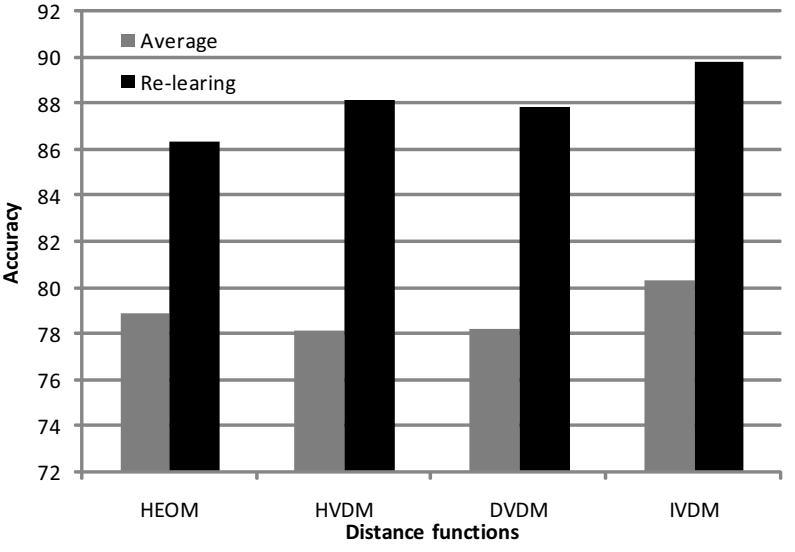


Fig. 3. Classification accuracy by relearning

3.3 Combining Ensemble Computation

To improve the accuracy, further in the classification, the operation of classifiers as a decision committee, is proposed. A committee as the final classifier, here, is composed of ensemble classifiers as committee members, each of which makes its own classifications that are combined to create a single classification result of the whole committee. An algorithm flow of the ensemble combining is shown in Fig. 4. In this study, the 10-fold cross-validation method is applied, which implies the data is divided 10 subsets. Then, the 9 subsets are training data and the remaining one subset becomes the testing data. Thus, the ensemble computations are carried out for 10 times.

The values in the columns in Fig. 4(a), show the classes of instances by ensemble computations. The classes are summed and voted as shown in Fig. 4(b); 3 for the class 0, 7 for the class 1 and 0 for the class 2. Then, the class 1 is determined for the instance 1 as shown in Fig. 4(c) are summed as shown in Table 1, as an example. The data is composed 16 instances in Table 1. The ensemble computations with 10 times are combined in the classification of the 0 class with 3 times, and that of the 1 class with 7 times. Thus, the final classification in the instance 1, becomes the 1 class by voting the majority of 7 times of 1. The class of the instance 1, is given as the class 1 in the data which is shown in the column, "correct" in Table 1. Then, the there is no error between the classified result, class 1, and the given class 1 in the data. But, in the instance 5, the combining computations with 10 times, are combined in the classification of the 0 class with 6 times and that of the 1 class with 4 times. Thus the final classification in the instance 5, becomes the 0 class by voting the majority of 6 times of 0. In this case, we have error $\times$ in Table 2.

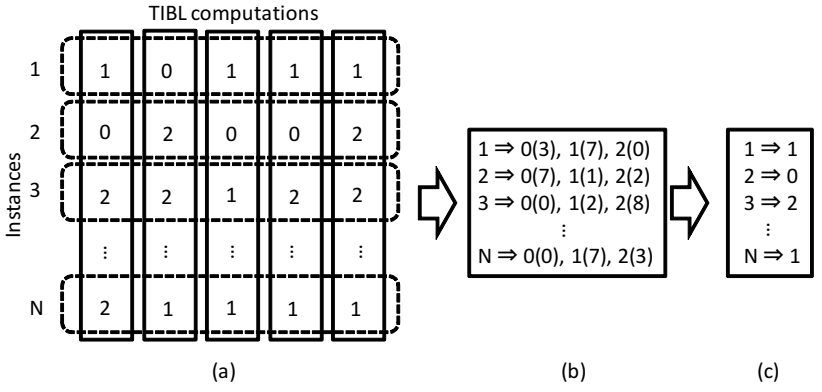


Fig. 4. Flow of combining ensemble computation algorithm

Table 2. Example of combining ensemble computation in Promot

instance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0
1	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
result	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0
correct	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
error					×					×	×					

3.4 Experimental Results for Relearning and Combining Ensemble Computations

For evaluating the classification generalization accuracy of our algorithm with the relearning and combining ensemble computation, was tested on 7 benchmark dataset from the UCI Machine Learning Repository [8]. The 10-fold cross-validation[1, 4] was applied. That is, the whole dataset is partitioned into ten subsets. Nine of the subsets are used as training set, and the 10th is used as the test set, and this process is repeated ten times. Then, classification accuracy is taken as the average of these ten runs. In our experiment, set parameters k = 3 and use four functions. The combining ensemble processing is verified by comparing the method with relearning process only and the method with relearning and combining ensemble computation by applying one distance function. Then, the generalization accuracy is shown in Table 3. The highest accuracy achieved for each dataset is shown in bold type as shown in Table 3. The method with the relearning and combining ensemble computation, is superior than only the relearning. The results by the combination method with the relearning and combining ensemble computations, which is called here, Combining, are shown in Table 3. The relearning results are also shown in Table 3 for the comparison. The bold numerals in Table 3, show the higher accuracy value between the results by only the relearning process and by the combination of the relearning and the combining en- semble

Table 3. Experimental accuracy results by method with relearning and combining ensemble computation

DataSet	HEOM		HVDM	
	Re-learning	Combining	Re-learning	Combining
Breast	95.51	**97.77**	96.84	**99.16**
Bridges	72.07	**76.00**	72.16	**82.89**
Flag	70.27	**71.62**	82.11	**86.14**
Glass	79.57	**83.18**	85.86	**95.28**
Heart	87.63	**89.45**	87.85	**97.02**
Hearttlb	92.00	**93.09**	84.71	**88.67**
Heartswi	97.77	**97.78**	97.00	**98.18**
Hepatit	90.05	**92.10**	89.43	**95.00**
Promot	90.54	**98.20**	**97.61**	97.38
Average	**86.38**	**88.80**	**88.19**	**94.44**
DataSet	DVDM		IVDM	
	Re-learning	Combining	Re-learning	Combining
Breast	97.87	**99.00**	97.42	**98.02**
Bridges	81.14	**91.81**	77.54	**92.61**
Flag	80.37	**95.57**	81.03	**95.14**
Glass	66.19	**80.85**	89.95	**94.55**
Heart	88.52	**95.87**	87.11	**94.28**
Hearttlb	90.05	**93.30**	90.59	**96.65**
Heartswi	97.85	**99.09**	**98.49**	98.18
Hepatit	91.35	**98.00**	90.66	**97.05**
Promot	97.63	**99.38**	**95.52**	95.00
Average	**87.89**	**94.76**	**89.81**	**95.80**

computation in using respective distance function. The distance function, IVDM shows higher accuracy values compared to other distance functions, HEOM, HVDM, and DVDM. In the dataset, “Heartswi” and “Promo” shows the higher accuracy value than Combining by the distances, HVDM and IVDM. These dataset has higher classification accuracy, in their dataset property.

So, their higher accuracy shows almost same in the Relearning and Combining. Thus, the combination of the relearning and combining ensemble computation, shows the superiority of the classification accuracy in Table 3, by comparison with the relearning process only. This shows the effectiveness of the combining ensemble computations in using the genetic algorithm.

4 Comparison with Other Methods

Many inductive earning algorithms has been proposed for classification problems. For example, ID3, C4.5, k-Nearest Neighbor, Na?ve-Bayes, IB, T2, Neural-network, association rules etc are developed. Among these developed methods, the results by well known methods are shown in Table 4 as C4.5(using Tree, Pruned-Tree, and Rule methods), IB(Instance-based learning algorithms, IB1, IB2) and Bayes (na?ve Bayesian

Table 4. Comparison with other conventional methods

DataSet	C4.5			IB		Bayes	BP	Re-learning	Combine
	Tree	P-Tree	Rule	IB1	IB2				
Breast	92.90	93.90	95.30	95.90	92.30	93.60	96.30	**97.41**	**98.49**
Bridges	68.00	65.30	59.50	53.80	45.60	66.10	67.60	**75.73**	**98.49**
Flag	59.20	61.30	60.70	63.80	59.80	52.50	58.20	**78.45**	**85.83**
Glass	68.30	68.80	68.60	70.00	68.80	71.80	68.70	**80.39**	**88.46**
Heart	73.30	72.10	80.00	76.20	68.90	75.60	82.60	**87.80**	**94.15**
Hematite	77.70	77.50	78.80	80.00	67.80	57.50	68.50	**90.37**	**95.65**
Promot	73.30	71.90	79.10	81.50	72.90	78.20	87.90	**95.33**	**97.49**
Average	72.97	72.56	74.01	74.53	68.47	71.36	75.59	**88.07**	**93.45**

classifier and BP(Back Propagation Neural Network) on the UCI repository data. Among the conventional methods, IB1 method shows better classification accuracy. The IB1 is a simple nearest neighbor classifier with k=1. The proposed methods here with the relearning computation and the combination of the relearning and the ensemble computations, are also presented in Table 4, for the comparison of the classification accuracy. The bold numerals in both Relearning method and Combine method, in Table 4, show the higher accuracy values compared with other conventional methods. The Combine (which is the combination of the relearning and ensemble computations) shows the highest accuracy in Table 4.

5 Conclusions

Among the conventional developed classification algorithms, the instance-based learning algorithm using k-nearest neighbor is useful one. But, only the k-nearest neighbor algorithm has several weaknesses in the application. To cope with these problems, improvements for the basic k-nearest neighbor method, are needed in the classification problems. In this paper, a relearning and combining ensemble computations, is proposed by applying useful distance functions. The classification accuracy by the proposed method of relearning and ensemble computation, shows the superiority compared with other conventional methods.

Combining different distance approach will be useful for the further improvement in the classification accuracy.

References

1. Wilson, D.R., Martinez, T.R.: An Integrated Instance-Based Learning Algorithm. Computer Intelligence 16(1), 1–28 (2000)
2. Bao, Y., Tsuchiya, E., Ishii, N., Du, X.: Classification by Instance-Based Learning Algorithm. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) IDEAL 2005. LNCS, vol. 3578, pp. 133–140. Springer, Heidelberg (2005)
3. Bao, Y., Ishii, N., Du, X.: A Tolerant Instance-Based Learning Algorithm. In: Dosch, W., Lee, R.Y., Wu, C. (eds.) SERA 2004. LNCS, vol. 3647, pp. 14–22. Springer, Heidelberg (2006)

4. Wilson, D.R., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 3–21 (1997)
5. Witten, I.H., Frank, E.: Data Mining Practical Learning Tools and Techniques. Morgan Kaufman, USA (2005)
6. Bay, S.D.: Nearest neighbor classification from multiple feature subsets. Intelligent Data Analysis 3, 191–209 (1999)
7. Kaneko, S., Igarashi, S.: Combining Multiple k-Neighbor Classifiers Using Feature Combinations. IEICE TRANSACTIONS on Information and Systems l.2(3), 23–31 (2000)
8. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases, Irvine, CA: University of California Irvine. In: Department of Information and Computer Science (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
9. Pawlak, Z.: "Rough Sets". Kluwer Academic Publishers, Dordrecht (1991)
10. Pawlak, Z.: Decision Networks. Rough Sets and Current Trends in Computing 2004. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 1–7. Springer, Heidelberg (2004)
11. Yamada, T., Yamashita, K., Ishii, N.: Text Classification by Combining Different Distance Functions with Weights. In: Proc. of SNPD 2006, pp. 85–90. IEEE Computer Society, Los Alamitos (2006)

Analysis of Agents' Cooperation in RoboCupRescue Simulation

Kazunori Iwata[1], Nobuhiro Ito[2], Kuniyoshi Toda[3], and Naohiro Ishii[4]

[1] Dept. of Business Administration, Aichi University
370 Kurozasa, Miyoshi-cho, Nishikamo-gun, Aichi, 470-0296, Japan
kazunori@vega.aichi-u.ac.jp
[2] Dept. of Applied Information Science, Aichi Institute of Technology
1247 Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392, Japan
n-ito@aitech.ac.jp
[3] Dept. of Compute Science and Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555, Japan
agent-staff@phaser.elcom.nitech.ac.jp
[4] Dept. of Applied Information Science, Aichi Institute of Technology
1247 Yachigusa, Yakusa-cho, Toyota, Aichi, 470-0392, Japan
ishii@aitech.ac.jp

Abstract. In this paper, we present a method to evaluate agents' cooperation in a Multi-Agent System (MAS). In the MAS research area, a MAS can be evaluated like according to total points, games won, targets reached and so on. The results do not, however, give an evaluation of the agents' cooperation due to the difficulty of investigating and evaluating the role of each agent, but instead give an total evaluation of the entire MAS. Against this background, we focus on the RoboCupRescue Simulation. The RoboCupRescue Simulation is used as the testbed environment and simulates, on a network of computers, a great earthquake and various kinds of disaster-relief activities by multi-agents in a virtual city. The evaluation of the MAS in this case is given by a "score", which is defined to reflect the disaster damage and does not give an evaluation of the agents' cooperation. Hence, we investigate a new indicator that can evaluate the agents' cooperation in the system. We consider and define 5 kinds of agents' cooperation dependent on Joint Intention Theory, Joint Responsibility Theory, the COM-MTDP Model and Coordination Theory. We also define cooperation in the RoboCupRescue Simulation by separating the definition into 10 definitions. Finally, we analyze the results of the elimination round in the RoboCupRescue Simulation League 2006 and consider the results of this analysis.

Keywords: Multi-Agent Systems, Cooperation, RoboCupRescue.

1 Introduction

In this paper, we present a method to evaluate the agents' cooperation in a Multi-Agent System (MAS). In the MAS research area, a MAS can be evaluated based on total points, games won, targets reached and so on. The results do not, however, give an evaluation of the agents' cooperation due to the difficulty of investigating and evaluating the role of each agent, but instead give an evaluation of the entire MAS. In a multi-agent simulation, for instance, the MAS indicates the best result, but the reason for the result is not obvious nor which which agent plays the most important role in obtaining the

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 227–236, 2008.
springerlink.com

result. Against this background, we focus on the RoboCupRescue Simulation[1, 5, 7]. The RoboCupRescue Simulation is used as the testbed environment and simulates, on a network of computers, a great earthquake and various kinds of disaster-relief activities by multi-agents in a virtual city. The evaluation of the MAS in this case is given by a "score", which is defined to reflect the disaster damage and does not give an evaluation of the agents' cooperation. Hence, we investigate a new indicator to evaluate the agents' cooperation in the system. For this purpose, we consider cooperation dependent on Joint Intention Theory, Joint Responsibility Theory, the COM-MTDP Model and Coordination Theory, and define the agents' cooperation in the RoboCupRescue Simulation. Finally, we analyze the RoboCupRescue Simulation League 2006 and consider the results of this analysis.

2 RoboCupRescue Simulation

The RoboCupRescue Simulation is used as the testbed environment and, on a network of computers, simulates a great earthquake and various kinds of disaster-relief activities by multi-agents in a virtual city. The agents in the RoboCupRescue Simulation are the "Fire Brigade (FB)", "Ambulance Team (AT)" and "Police Force (PF)". The FB extinguishes fires while the AT rescues a buried civilian in a building, and the PF clears a blocked road. Then, the teams (study groups from colleges or companies) that belong to the RoboCupRescue Simulation League develop these three agents. The "score" which is the evaluation of the RoboCupRescue Simulation League, is defined by Eq. (1) and reflects disaster damage. However, it does not give an evaluation of any agent.

$$P = \left(n_{al} + \frac{h_{al}}{h_{aa}} \right) \times \sqrt{\frac{S_{NoBurned}}{S_{all}}} \tag{1}$$

n_{al} : Number of surviving agents.
h_{al} : Sum of the surviving agents' HP, which is short for "Health Point", and means the vitality of an agent in the RoboCupRescue Simulation.
h_{aa} : Sum of all agents' HP.
$S_{NoBurned}$: Sum of the architectural areas of the flameless buildings.
S_{all} : Sum of the architectural areas of all buildings.

3 Agents' Cooperation

By considering agent's cooperation, we investigate a new indicator to be used to evaluate each agent in the system. In this section, we explain agents' cooperation as used in this paper. First, we introduce the definition of agents' cooperation from related works, and then give our own definition thereof.

3.1 Teamwork Theories

In the teamwork theories, cooperation is achieved by forming teams, which are groups of agents the same goals. The fundamental teamwork theory is Joint Intention Theory[2], while more practical theories are Joint Responsibility Theory [3] and the COM-MTDP (COMmunicative Multi-agent Team Decision Problem) Model[6].

3.1.1 Joint Intention Theory

Joint Intention Theory aims to create valid teamwork. In this theory, agents achieve cooperation through a joint intention, namely that "the members of the team have the same goals and act to achieve them", and by committing to it and to each other. An agent that commits to the joint intention and to other agents in a team takes responsibility for the team activities. Here responsibility means that if the agent wishes to contract out of the team, it must be approved by all members. This responsibility prevents any inconsistency arising when some members act on other intentions despite having accepted the commitment.

3.1.2 Joint Responsibility Theory

Joint Responsibility Theory is based of Joint Intention Theory. In this theory, an agent selects a goal and decides whether he needs other agents' cooperation or not. If he decides to involve other agents, they establish a team.

3.1.3 COM-MTDP Model

The COM-MTDP Model extends the team creation procedure using communication. This model has the following characteristics:

- Quantitatively evaluating teamwork.
- Analyzing the calculation involved team creation.
- Applying it to any other teamwork theories.

The COM-MTDP model deals with rewards that are given for agents' activities.

3.2 Coordination Theory

Coordination Theory focuses on the interdisciplinary study of coordination[4]. Research in this area uses and extends ideas about coordination from disciplines such as computer science organization theory, operations research, economics, linguistics, and psychology. The important research related to multi-agent systems is that a cooperation can be seen as the process of managing dependencies between agents' activities. Dependencies are divided into the following two types:

Task Dependencies : an agent ($agent_1$) cannot begin his task before finishing another agent's task ($task_o$), because $agent_1$ does not have the ability to complete $task_o$.

Task / Subtask Dependencies : if a task is divided into several subtasks, all the subtasks must be completed to complete the task.

3.3 Definition of Agents' Cooperation

First, we define agents' cooperation based on the Joint Responsibility Theory (Sect. 3.1.2). According to this theory, agents establish a team if one of them requires another agent's help, because the agent is not able to complete the task. The reasons why an agent cannot carry out a task by himself are:

1. The task is too large for him to complete; more precisely, the task needs to be carried out by several agents.
2. The agent does not have the ability to carry out the task.

In the RoboCupRescue Simulation, for instance, the AT would like to carry out the task of rescuing a buried civilian in a building. If the civilian is buried deeply, an agent cannot dig out the civilian alone (an example of case 1 above). Furthermore, if the AT comes across a fire, they cannot extinguish it without the help of the FB (illustrating case 2 above). Similarly, the FB's task is to extinguish fires while the PF's task is to clear blocked roads. The cooperation required in case 1 is defined as *JR1* and that in case 2 as *JR2*.

Definition 1. Cooperation *JR1*
Cooperation *JR1* implies the following situation:

> Owing to the size of a task being too large, an agent requests help from other agents, through communication and they heed the request.

Definition 2. Cooperation *JR2*
Cooperation *JR2* implies the following situation:

> Because an agent does not have the ability to complete a task, he requests other agents to help him through communication and they heed the request.

Next, we define agents' cooperation based on the COM-MTDP Model (Sect. 3.1.3). In this model, the more agents cooperate with each other, the more rewards they receive. Hence, the team that receives the highest number of rewards, cooperated the most. In this paper, we assume that the shorter the time to finish a task, the more rewards a team receives. We also assume a cost of communication among agents, because they must communicate with each other to cooperate. Therefore, cooperation in the COM-MTDP Model (*CMTDP*) is defined as follows:

Definition 3. Cooperation *CMTDP*
Cooperation *CMTDP* implies the following situation:

> More than two agents complete a task as quickly as possible by considering communication costs.

Next, we define agents' cooperation based on Coordination Theory (Sect. 3.2). In this theory, cooperation can be seen as the process of managing dependencies between agents' activities.

Cooperation for "Task Dependencies" is defined as *CT1* and that for "Task / Subtask Dependencies" as *CT2*.

Definition 4. Cooperation *CT1*
Cooperation *CT1* implies that:

> The process manages "Task Dependencies", and besides, the time for completing a task is considered.

Definition 5. Cooperation *CT2*
Cooperation *CT2* implies that:

> The process manages "Task / Subtask Dependencies", and besides, the time for completing a task is considered. Here, subtasks do not have "Task Dependencies" (related to CT1) and each subtask can be completed by a single agent (and cannot be further subdivided).

The key points of these definitions are given in Table 1. The items in the table are as follows:

Homogeneous : If a definition deals with agents that have the same ability, the entry is ○, else the entry is ×.
Heterogeneous : If a definition deals with agents that have different abilities, the entry is ○, else the entry is ×.
Size of Tasks : If a definition focuses on the sizes of tasks, the entry is ○, otherwise the entry is ×.
Times : If a definition focuses on times to complete a task, the entry is ○, otherwise the entry is ×.
Communication : If a definition focuses on agents' communication, the entry is ○, otherwise the entry is ×.
Completing Tasks : If a definition focuses on completion of related tasks, the entry is ○, otherwise the entry is ×.

Table 1. Features of Agents' Cooperation for each Definition

	JR1	*JR2*	*CMTDP*	*CT1*	*CT2*
Homogeneous	○	×	×	×	○
Heterogeneous	×	○	○	○	○
Size of Tasks	○	×	×	×	×
Times	×	×	○	○	○
Communication	○	○	○	×	○
Completing Tasks	×	×	○	○	○

4 Cooperation in RoboCupRescue Simulation

Cooperation in the RoboCupRescue Simulation is dependent on the definitions from Sect. 3 being defined as follows:

Definition 6. *JR1* in RoboCupRescue Simulation

FB_JR1

FB_JR1 is calculated as the number of FBs that extinguish a fire in the same building.

AT_JR1

AT_JR1 is calculated as the number of ATs that rescue a buried civilian where one AT cannot rescue the civilian by itself.

Definition 7. *JR2* in RoboCupRescue Simulation

FB-PF_JR2

FB-PF_JR2 is calculated as the number of PF activities that are requested by FBs.

AT-PF_JR2

AT-PF_JR2 is calculated as the number of PF activities that are requested by ATs.

Definition 8. *CMTDP* in RoboCupRescue Simulation

FB-PF_CMTDP

FB-PF_CMTDP is calculated as the number of PF activities that are requested by FBs within 10 time-steps after a fire breaks out.

AT-PF_CMTDP

AT-PF_CMTDP is calculated as the number of PF activities that are requested by ATs within 60 time-steps after a civilian is buried.

Definition 9. *CT1* in RoboCupRescue Simulation

FB-PF_CT1

FB-PF_CT1 is calculated as the number of PF activities that clear blocked roads hindering FBs.

AT-PF_CT1

AT-PF_CT1 is calculated as the number of PF activities that clear blocked roads hindering ATs.

Definition 10. *CT2* in RoboCupRescue Simulation

FB_CT2

FB_CT2 is the remaining simulation steps required for FBs to extinguish all fires in burning buildings.

AT_CT2

AT_CT2 is the remaining simulations steps required for ATs to rescue all buried civilians.

Table 2 shows which agents' cooperation the simulation results denote and is based on Table 1. For instance, *FB_JR1* only has "FB" as the kind of FB, which means that *FB_JR1* can evaluate cooperation between FBs only. Furthermore, *FB_JR2* has "FB" as the kind of PF, which means that *FB_JR2* can evaluate the cooperation of the PFs with FBs only. Similarly, *FB_CT2* has "FB, PF " as the kind of FB, which means that *FB_CT2* can evaluate the cooperation of the FBs with FBs or PFs. Therefore, to judge the cooperation of FBs in a MAS, *FB_JR1* and *FB_CT2* must be considered. The cooperation of ATs is judged similarly, while to judge cooperation between FBs or ATs and PFs in a MAS, *JR2*s, *CMTDP*s, *CT1*s and *CT2*s must be considered.

Table 2. Candidates for Cooperation in RoboCupRescue Simulation

Kinds	*JR1*		*JR2*		*CMTDP*		*CT1*		*CT2*	
	FB_ JR1	*AT_ JR1*	*FB-PF_ JR2*	*AT-PF_ JR2*	*FB-PF_ CMTDP*	*AT-PF_ CMTDP*	*FB-PF_ CT1*	*AT-PF_ CT1*	*FB_ CT2*	*AT_ CT2*
FB	FB	×	×	×	PF	×	×	×	FB, PF	×
AT	×	AT	×	×	×	PF	×	×	×	AT, PF
PF	×	×	FB	AT	FB	AT	FB	AT	FB	AT

5 Analysis and Considerations

5.1 Analyzing Simulation Results

We now analyze the results of the RoboCupRescue Simulation League 2006 according to the definitions given in Sect. 4. In the 2006 league, there were 19 teams and each team played 13 competitions in the elimination round. Thus there arr 247 simulations. Here, one competition means that the agents play for 300 time-steps on a map of a stricken area. The results of the analysis are shown in Table 3 and the ranks of the values in Table 3 are given in Table 4.

5.2 Considering the Results of the Analysis

5.2.1 Considering the Cooperation between FBs

The *FB_JR1* values in Table 4 indicate that the cooperation between FBs in Kosar, CSU_YunLu, Impossibles, IUST, and SBCe_Saviour is better than that for other teams.

Table 3. Results of Analyzing RoboCupRescue Simulation

Team	*FB_JR1*	*AT_JR1*	*FB-PF_JR2*	*AT-PF_JR2*	*FB-PF_CMTDP*	*AT-PF_CMTDP*	*FB-PF_CT1*	*AT-PF_CT1*	*FB_CT2*	*AT_CT2*
Aladdin	760	72	1238	1190	749	568	4064	1854	464	0
BAM	592	173	1464	960	1179	423	4054	1682	160	0
CSU_YunLu	3349	90	661	518	757	535	4313	1885	335	0
DAMAS	1522	54	1717	1116	1272	542	4409	1889	284	0
FCPortugal	1048	130	522	388	914	368	4316	1911	183	0
Hinomiyagura	351	7	1810	1921	1159	230	1435	1043	0	0
Impossibles	3077	9	819	669	679	429	4375	1832	633	134
Incredibles	1742	30	776	844	796	594	4240	1843	493	0
Itandroids	1358	7	525	479	641	350	4374	1822	170	0
IUST	3215	10	544	286	600	613	4080	1885	683	0
Kosar	3671	148	634	581	823	693	4371	1811	458	0
Kshitij	987	46	862	642	552	480	4430	1799	144	0
MRL	1633	98	794	695	837	582	4503	1941	781	0
NITRescue06	1291	0	1029	286	815	201	3814	1170	164	0
Persia	1161	181	1022	1099	770	673	4061	1579	376	0
Poseidon	2184	8	528	272	741	351	3254	1398	801	83
RoboAKUT	1434	286	861	531	595	475	4419	1832	274	0
S.O.S.	1536	187	679	438	847	567	4429	1897	434	36
SBCe_Saviour	2901	0	549	390	533	428	4435	1952	727	0

Table 4. Ranks of Values in Table 3

Team	*FB_JR1*	*AT_JR1*	*FB-PF_JR2*	*AT-PF_JR2*	*FB-PF_CMTDP*	*AT-PF_CMTDP*	*FB-PF_CT1*	*AT-PF_CT1*	*FB_CT2*	*AT_CT2*
Aladdin	17	9	4	2	12	6	14	8	7	4
BAM	18	4	3	5	2	14	16	15	17	4
CSU_YunLu	2	8	13	12	11	9	11	6	12	4
DAMAS	10	10	2	3	1	8	6	5	13	4
FCPortugal	15	6	19	16	4	15	10	3	14	4
Hinomiyagura	19	16	1	1	3	18	19	19	19	4
Impossibles	4	15	9	8	14	12	7	10	4	4
Incredibles	7	12	11	6	9	4	12	9	5	4
Itandroids	11	16	18	13	15	17	8	12	15	4
IUST	3	14	16	18	16	3	13	6	6	1
Kosar	1	5	14	10	7	1	9	13	8	4
Kshitij	16	11	7	9	18	10	3	14	18	4
MRL	8	7	10	7	6	5	1	2	2	4
NITRescue06	12	18	5	17	8	19	17	18	16	4
Persia	14	3	6	4	10	2	15	16	10	4
Poseidon	6	13	17	19	13	16	18	17	1	2
RoboAKUT	13	1	8	11	17	11	5	10	11	4
S.O.S.	9	2	12	14	5	7	4	4	9	3
SBCe_Saviour	5	18	15	15	19	13	2	1	3	4

However, the *FB_CT2* values for CSU_YunLu, IUST and Kosar are not good. Judging in a comprehensive manner, the Impossibles and SBCe_Saviour teams produced the most efficient cooperation between FBs.

5.2.2 Considering the Cooperation between ATs

The *AT_JR1* values in Table 4 indicate that the cooperation between ATs in BAM, Kosar, Persia, RoboAKUT, and S.O.S. is better than that of other teams. The values of *AT_CT2* for S.O.S is also better than that for the other teams, most of which do not achieve good results. Hence, the S.O.S. team produced the most efficient cooperation between ATs.

5.2.3 Considering the Cooperation between FBs or ATs and PFs

Most of the values of *JR2* and *CMTDP* for DAMAS, Incredibles, Kosar, MRL, Persia and S.O.S. are better than those for the other teams. However, the values of *CT1* for Incredibles, Kosar and Persia are not good. Hence, the DAMAS, MRL and S.O.S. teams performed most efficiently in terms of PFs' cooperation with FBs or ATs. Considering the cooperation between PFs and FBs the results are given in more detail below.

- Aladdin produced efficient cooperation between PFs and FBs, because *FB-PF_JR2* and *FB_CT2* represent good results despite the not so good results of *FB_JR1*.
- MRL produced efficient cooperation between PFs and FBs, because *FB-PF_JR2*, *FB-PF_CT1* and *FB_CT2* represent good results despite the not so good results of *FB_JR1*.
- In contrast with the above, the results of *FB-PF_JR2* and *FB-PF_CMTDP* for NITRescue06 show that the PFs cooperate with FBs, but that the FBs do not utilize the cooperation, because the result of *FB_CT2* is worse than that of *FB_JR1*.

The cooperation between PFs and ATs is also presented in more detail below.

- IUST produced efficient cooperation between PFs and ATs, because *AT-PF_CMTDP* and *AT-PF_CT1* represent good results despite the not so good results of *AT_JR1*.
- In contrast with the above, most results of *AT_JR1*, *AT-PF_JR2* and *AT-PF_CMTDP* for Kosar and Persia show that the PFs and ATs cooperated with each other, but that they did not cooperate efficiently, because the results of *AT_CT2* are the worst.

6 Conclusion and Future Work

In this paper, we have explained how to evaluate agents' cooperation in the RoboCupRescue Simulation. The evaluation of a RoboCupRescue Simulation is given as a "score", which is defined to reflect disaster damage and does not include an evaluation of each agent. Hence, we have considered and defined 5 kinds of agents' cooperation dependent on Joint Intention Theory, Joint Responsibility Theory, the COMMTDP Model and Coordination Theory. In addition, we have defined cooperation in the RoboCupRescue Simulation by separating the definition into 10 definitions. Finally, we have analyzed the RoboCupRescue Simulation League 2006 and considered the results of the analysis. This has shown that using our definitions we are able to evaluate agents' cooperation in the RoboCupRescue Simulation. Our future work is aimed at

accumulating and analyzing data regarding agents' communication as the results of the RoboCupRescue Simulation League do not include any results of communication among agents, despite, communication being an important factor in analyzing agents' cooperation.

Acknowledgement

This research was supported by Aichi University Grant C-146, by the Hori Information Science Promotion Foundation and by Grant-in-Aid for Scientific Research (No. 19700032) awarded from the Japan Society for the Promotion of Science.

References

1. Robocup rescue home, http://www.robocuprescue.org/
2. Cohen, P.R., Levesque, H.J.: Teamwork. Nous 25, 487–521 (1991)
3. Jenings, N.R.: Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. Artificial Intelligence 75(2), 87–240 (1995)
4. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM 26(1), 87–119 (1994)
5. Morimoto, T.: edited by RoboCupRescue Technical Commitee: How to develop a robocuprescue agent, http://ne.cs.uec.ac.jp/~morimoto/rescue/manual/index.html
6. Pynadath, D.V., Tambe, M.: Multiagent teamwork: Analyzing the optimality and complexity key theories and models. In: AAMS, pp. 873–880 (2002)
7. Tadokoro, S., Kitano, H.: RoboCup-Rescue Technical Committee, The RoboCup Federation. In: RoboCup Japanese National Comittee (eds.) RoboCup Rescue (in Japanese), Kyoritsu Shuppan Co. Lt. (2000)

A Data Mining Approach for Predicting Reliable Path for Congestion Free Routing Using Self-motivated Neural Network

B. Chandra Mohan, R. Sandeep, and D. Sridharan

Anna University, Chennai, India
abc@cs.annauniv.edu

Summary. Congestion in computer networks is a significant problem due to the growth of networks and increased link speeds. Now it is common to see internet gateway drops 10% of the incoming packets because of local buffer overflows. An optimal solution for this problem is Predicting congestion free path(s) by learning the dynamic characteristics of networks and its topology. The factors that influence the prediction of such path(s) have the characteristics viz., dynamic, non-linear, incertitude, etc., which make traditional data mining approach, like neural prediction have to process a large amount of convoluted data. In this paper, we proposed prediction model rather than mathematical model for finding congestion free path(s). We introduced a self-motivated learning in the training phase of an improved functional link feed-forward neural network for predicting reliable path that offer congestion free path(s).

1 Introduction

In October of '86, the Internet had the first of what became a series of congestion collapses. During this period, the data throughput from LBL to UC Berkeley [1] (sites separated by 400 yards and two IMP hops) dropped from 32 Kbps to 40 bps (99.88% packet drop). The following Table 1 shows the average response time and average packet loss of each continent on 08th January of 2008 at 20:00 IST. It is concrete from this experimental result [2] that the Average packet loss during even off peak hour is 14% and on peak hour is 25%. Due to this huge packet loss, the entire network efficiency has come down drastically. So, congestion control and congestion avoidance become most important problems in data communication. In this paper, we proposed an approach for finding reliable path which provides comparatively lesser average response time and lesser packet drops.

In studies of communication and computer networks, reliability has been defined in a number of different ways. A network has been defined to be operational in the presence of failures provided communication paths exist between certain pairs of nodes. Alternatively, a network has been considered to be operational in the presence of failures if every node could communicate with a certain percentage of the other nodes. However, these definitions would be more meaningful if they quantitatively reflected the traffic-carrying capacity of the network in the presence of failures. Based on several of the definitions of acceptable network operation given previously, a number of deterministic and probabilistic reliability criteria have been considered. Reliability is the probability that the time between failures in the network (including software, hardware, and communications) is greater than some positive value t (threshold).

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 237–246, 2008.
springerlink.com

Table 1. Average Response Time and Average Packet Loss of each Continent recorded in Jan 08, 2008 at 20.00 IST

Continent	Approx Time	Avg. Response Time (ms)	Avg. Packet Loss (%)
Asia	20:00	313	14 %
Australia	02:00	179	0 %
Europe	14:30	233	7 %
North America	07:30	92	2 %
South America	10:00	376	25 %

2 Mathematical Model for Congestion Control

A more general reliability criterion than the edge connectivity or cohesion of a graph, introduced by Boesch [3], is the minimum number of edges that must be removed from a graph in order to isolate any sub graph of *m* nodes from the rest of the graph. This quantity, denoted by $\delta(m)$, corresponds to the minimum number of communication link failures that will isolate any set of *m* node(s) from the remaining ones. Based on this criterion, maximally reliable networks clearly correspond to graphs for which $'\delta(m)$ is as large as possible for all values of $s(n)$. It is to be noted that this criterion is only meaningful as a computer-network reliability measure if the probability of communication-link failures is much greater than the probability of node(s) breakdowns in the network. On the other hand, if it were more likely for node(s) to fail than for communication links to fail, we could use a criterion analogous to $\delta(m)$ for the nodes of a graph, denoted by $\delta^*(m)$. In the design of maximally reliable computer networks based on the realization of $s(m)$ optimal graphs, only partial results are available.

More mathematical model was developed by many researchers in the past few decades. All the models represent feasible solution based on few network metrics. If we apply this model for real time situations, the performance decreases from the theoretical value due to the non-static nature of network parameters. And any one or more parameter may influence the current situation and it may not be influenced to some other situation, this impulsive nature of network not suitable for mathematical model. Identifying a reliable path from this impulsive nature is possible through the data mining approach. Here, we proposed data mining approach for predicting reliable path from all observational network data set, in order to control and avoid congestion.

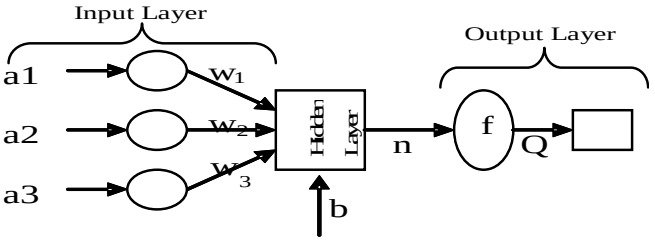


Fig. 1. General Structure of ANN

3 From Theory to Application (Using Artificial Neural Network Model)

Unlike Von Neumann machines based processing/memory abstraction of human information processing; neural networks are based on the parallel architecture of human brains. A neural network is a way of processing information that mimics the operation of the human brain, based on learned experience, which has advantages, notably speed and accuracy, over more mechanical methods. Artificial Neural Networks (ANN) has given appreciable result for hard real time problems like monitoring the state of aircraft engines, by monitoring vibration levels and sound, early warning of engine problems. British Rail have also been applied a similar application monitoring diesel engines for their rail route. Bellwether [4], explains more fully the concept of neural networks and their strengths and weaknesses, and examines how they might be used to address and solve problems and their potential value in a range of rail industry applications.

3.1 Algorithm for Feed Forward ANN (FFANN)

Initialize all weights and biases in network.

Set terminating condition (epoch & goal)

While terminating condition is not satisfied

For each training tuple - X in D

// propagate the inputs forward

For each input layer unit j

$O_j = I_j$;

// output is its actual input value.

For each hidden or output layer unit j

$I_j = E_j W_{ij} O_i + O_j$;

// compute the net input of unit j with respect to //the //previous layer, i

$O_j = 1/(1+e^{-Ij})$; }

// compute the output of each unit j

// Back propagate the errors.

For each unit j in the output layer

$Err_j = O_j (1-O_j) (T_j-O_j)$; //compute the error.

For each unit j in the hidden layers, from the last

to the first hidden layer.

$Err_j = O_j (1-O_j) E_k Err_k W_{jk}$;

//compute the error with respect

//to the next //higher layer, R

For each weight W_{ij} in network

$\Delta w_{ij} = (l) Err_j O_i$;

//weight increment

$W_{ij} = W_{ij} + \Delta w_{ij}$;

//weight update

For each bias Oj in network

$\Delta O_{ij} = (l) Err_j$;

//bias increment

$O_j = O_j + \Delta O_j$;

// bias update

End //while

where l is learning rate

4 Design and Implementation of ANN Model (Based on Data Mining Approach)

An association rule, which is an important field of study in data mining, is used widely for predicting underlying knowledge even from the secondary (or observation) data. Association rule mining helps us to identify set of constraints, rules, statements and frequent item sets regarding the user criteria in the large volume of data. Efficient mining of frequent item sets [5] is a fundamental problem for mining association rules. Many Apriori-like algorithms offer good performance for association rule mining.

Here training process of basic learning approach is applied for coordinating with association rules for redefining the knowledge extraction process. However, it is costly, especially for network data mining such as internet like huge connectivity. To

handle a large number of such data, leads large number of candidate sets and must scan the database repeatedly. As the amount of data increases, designing an efficient mining algorithm become increasingly urgent. The items should be given different weights to reflect their importance and the weighted association rules mining should be implemented. In our problem, each network parameter represents a weight and this weight is updated in the training phase based on network database. Now, applying variables with these updated weights give predicted value(s). Here, the predicted value means, a node sequence that represents a possible reliable path.

4.1 Weights Confirmed by the Neural Network

There are already many productions for weight analysis in mining association rules, but they are unfit for predicting reliability of telecommunication networks [6] due to huge data set and the corresponding candidate data generation. In this paper, we propose one neural cell network to confirm the data effectively. The model can be seen as the simplest kind of feed-forward neural network which contains three inputs, three link weights, one neuron and one output as shown in fig.1. In this neural network model, three parameters should be confirmed: (1) the ration of description for the inputs; (2) the link weights of the neural network; (3) the transfer function.

In our study, the input datasets are the report that contains data about packet drop and average response time in telecommunication network with pretreatment. Network Congestion attributes are made of many factors, of which influence the telecommunication network most should be chosen to form the inputs of the neural network including the node degree of the congestion equipment, the congestion level that may reflect the severity and the congestion type which can influence the network. The outputs of the neural network are the congestion weights that we need for choosing best path.

First the sample values from the experience of experts should be put into the input port. After training the link weights of the neural network, we will get the neural network model for confirming the congestion weights. Determine the transfer function is the final and crucial step in the neural network design process for process correlation analysis. In this design, according to the characteristics of congestion control data transmission, the saturated linear function is selected as follow,

$$f = \begin{cases} 0 & n < 0 \\ n & 0 \le n \le 1 \\ 1 & n > 1 \end{cases} \tag{1}$$

4.2 Analyzable Structured Neural Network (ASNN)

Fig. 2 shows structure of the ASNN [7], which has some network modules. The network module consists of two types of hidden units. One type of hidden units has connecting weights between only one group of related input units. The network module with this type of hidden units is called a sparse-connecting module. Another one has connecting weights between all input units. The network module with this type of hidden units is called an all-connecting module. The former type of hidden units

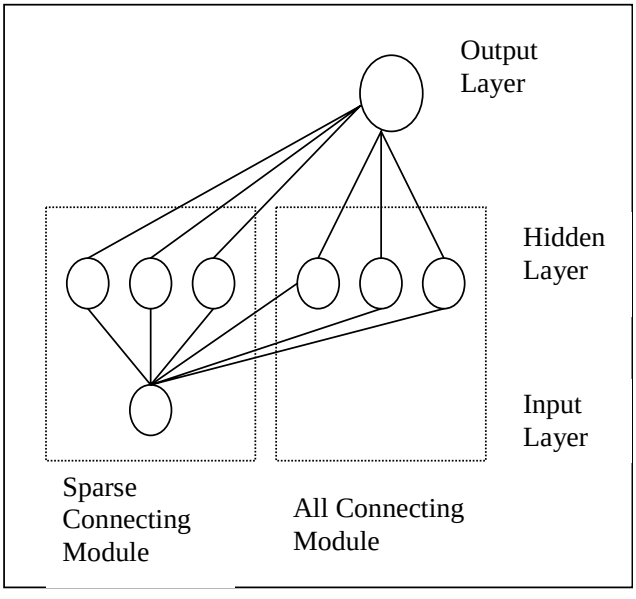


Fig. 2. Analyzable Structured Neural Network

allows to analyze each relation between a certain input data and a corresponding output data. The latter type of hidden units insure the performance of the neural network as same as the conventional ANN.

4.3 Efficient Methodology for Predicting Reliable Path Using Self Motivated ANN Model

Here we implemented an improved version of feed forward neural network, namely self-motivated functional link feed forward neural network. In our design we have

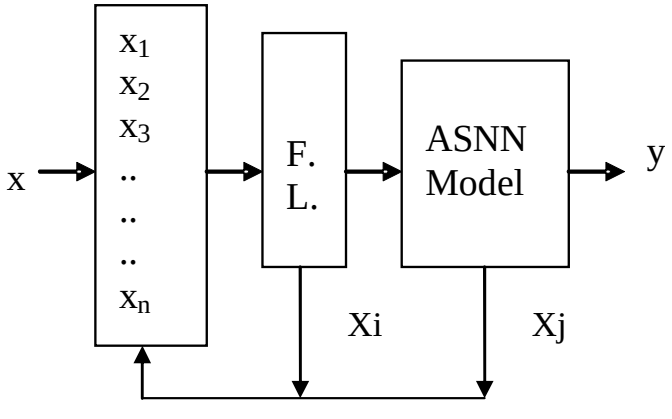


Fig. 3. Self-Motivated ANN with improved functional link architecture

additional input in order to train the neural network. This additional input plays vital role on training phase comparing to existing training and learning procedures. The architecture of our proposed self-motivated ANN with improved functional link architecture is shown in fig.3. Here X [X1, ..Xn] is an input (training tuples), Xi & Xj are additional inputs. Xi generated based on functional link and Xj generated based on output.

Learning algorithm lies heavily on the ordering structure of sample space with an alternative feed-forward and back propagation technique. The process of developing a neural network model for a particular application usually involves four basic stages. 1).A network developer selects a problem domain, such as theoretical, empirical, or applied interests (in our case, congestion control). 2).Network architecture is designed for capturing the underlying criteria from the problem domain. This architecture forms the configuration of the network including the number of units used, their organization into layers, learning parameters, and error tolerance (in our case, routing table of particular node and network parameters like average packet loss and average response time of possible routes). 3).Chosen architecture and a chosen task, a learning paradigm such as back propagation are applied to train the network and develop the inter-connection weights (in our case, we used ASNN Architecture). 4).Developer evaluates the trained network according to objective performance measures such as its ability to solve the specified task and its ability to predict the outcome of unseen cases.

Learning in neural networks represents one of their great capabilities. It takes place in by adjusting the connection weights between simple processing units (in our case, we applied self-motivated training).

The self-motivated neural network used the following mathematical model for learning when it is trained. Is it training or learning?, answer is, both. An unsupervised learning using the following mathematical model is implemented when it is in training phase. The need of such learning is important because of the dynamic characteristics of network parameters that include topology are highly not stable and it may alter it nature at any seconds or even millisecond. The historical data set of training data not enough to identify such an efficient and reliable path. And in some cases, storing such a huge data set in a middleware like router is not appreciable, so self learning from the set of sequence gives alternate solution to this problem.

5 Result and Analysis

The Table 2 shows the theoretical value of error in the hidden node and output node of first three iterations for both feed forward neural network and our self-motivated feed forward neural network with improved functional link techniques. The difference between traditional ANN and our proposed work is shown in the same (table 2). Based on these values it is concluded that the error in each hidden node and output node always less than traditional feed forward. So our methodology requires less no of epoch than the previous algorithm for achieving its goal.

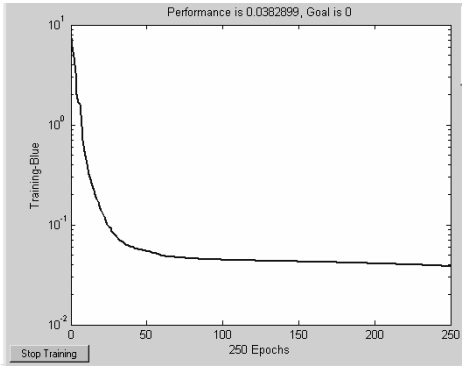
The Fig. 4 shows the various implementation results. We implemented Reliable Path finding from network data using the data mining approach based on

Table 2. Various ANN and its Error calculation of hidden and output node on I, II, and III iteration

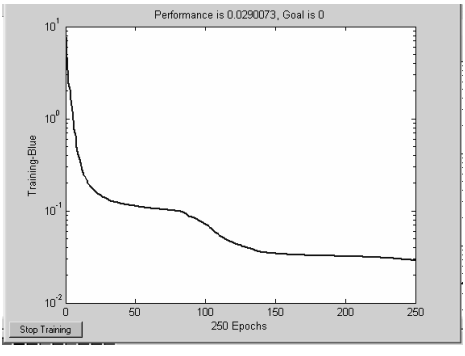
Feed Forward ANN			
Iteration	Error input1	Error input1	Error output2
I	-0.0101	-0.0066	0.1341
II	-0.0069	-0.0040	0.1209
III	-0.0052	-0.0022	0.1110
Self-motivated improved Functional Link ANN			
Iteration	Errorinput1	Error iput1	Error output2
I	-0.0072	-0.0006	0.1344
II	-0.0010	-0.0003	0.1162
III	-0.0070	0.0005	0.1093
Performance Improvement			
Iteration	Error input1	Error iput1	Error output2
I	-0.0029	-0.0061	-0.0003
II	-0.0042	-0.0020	-0.0053
III	-0.0001	-0.0011	0.0251

1).Traditional Feed Forward ANN and 2).Self-motivated ANN with improved functional link techniques using MatLab (Version 14). Fig 4.a) shows the result of error in the output node of Feed Forward ANN after training phase completed Fig.4.b) shows the result of error in the output node of our proposed work after training phase completed Fig.4.c) shows the result of real time error between actual and predicted values of both Feed forward ANN and our proposed work.

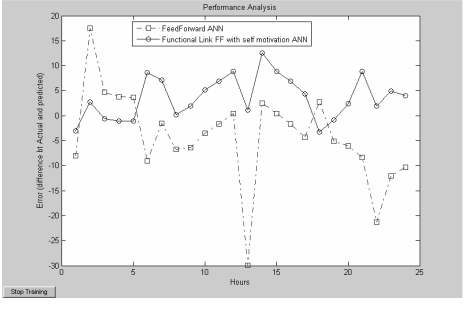
In the training phase of our implementation we gave epoch as 250 and goal as 0.01. In the traditional ANN the performance (error in output node) achieved as 0.0382899 (~ 4%). Where as in our proposed work, the error in the output node become comparatively less, 0.0290073 (~ 3%). The difference may be less but it gives best result when we predict the reliable path. Fig.4.C shows the Congestion in the reliable path of both traditional ANN and our proposed work. Congestion in the network path becomes less in the path predicted by traditional ANN comparing to the path that decided by the present routing protocol. For the same network, Congestion in the network path becomes less in the path predicted by our proposed ANN comparing to the path that predicted by traditional ANN.



a). Error in each iteration of Feed Forward ANN (epoch: 250, goal: 0.01, achieved: 0.0382899



b). Error in each iteration of self-motivated improved Functional Link ANN (epoch: 250, goal: 0.01, achieved: 0.0290073



c). Performance comparison between Feed Forward and self-motivated improved Functional Link ANN

Fig. 4. Result of our MatLab implementation

6 Conclusions and Future Works

Our proposed work, gives optimal solution for congestion control in order to identifying reliable path from network data base. So we suggested that Predicting reliable

path is an optimal solution for all real time networking problems comparing hop count shortest path routing protocol. Network resource is a crucial problem for implementing this congestion control mechanism. But the cost of resource in the current industry is very less, so all the way for congestion problems, our approach gives best solution.

This paper concentrates only on design and software perspective, the hardware perspective is not included in this paper. We suggested the enthusiastic reader of this article may implement it in the VLSI design of middleware.

References

[1] Jacobson, V.: Congestion Avoidance and Control. In: Proceedings of SIGCOMM 1988, Palo Alto, CA (August 1988)

[2] Real time traffic monitoring, http://www.internettrafficreport.com/details.htm

[3] Boesch, T., Thomas, R.E.: On Graphs of Invulnerable Communication Nets. IEEE Transactions On Circuit Theory 17(2), 183 (1970)

[4] Iizaka, T., Matsui, T., Fukuyama, Y.: A novel daily peak load forecasting method using analyzable structured neural network. In: IEEE/PES Transmission and Distribution Conference and Exhibition 2002. vol. 1, pp. 394–399 (October 2002)

[5] The use of neural networks in the rail industry (R&D Report), http://www.rssb.co.uk/pdf/reports/research/The%20use%20of%20neural%20networks%20in%20the%20rail%20industry.pdf

[6] Li, T., Li, X., Xiao, H.: An Effective Algorithm for Mining Weighted Association Rules in Telecommunication Networks. In: International Conference on Computational Intelligence and Security Workshops, pp. 425–428. IEEE computer society, Los Alamitos (2007)

[7] Chen, C.-H., Lin, C.-T., Lin, C.-J.: A Functional-Link-Based Fuzzy Neural Network for Temperature Control. In: Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence FOCI 2007, pp. 53–58 (2007)

A Formal Specification of UML Class and State Diagrams

Gongzhu Hu

Department of Computer Science
Central Michigan University
Mount Pleasant, Michigan, USA
hu1g@cmich.edu

Summary. UML has been the *de facto* specification language for software design, using the object-oriented approach in particular. However, components in UML are mostly presented as graphical diagrams that are informal and often lead to different interpretations. Although formal methods exist for UML specification, most of which are semi-formal or rely on another high-level language to describe the UML components. In this paper, we give a formal specification for the core UML, the class and state diagrams using the basic predicate logic and set notations. The goal is to provide a precise definition for object-oriented software.

1 Introduction

UML is the standard specification language for software design, including software components, their relationships and behavior. It is mostly used in the form of UML diagrams that are graphical representations of software designs. Although expressive and powerful, UML diagrams to certain degree are still ambiguous because of its informal nature, just as any informal approach to a design problem. It is always desirable to describe a design problem in a formal way so that the software components are precisely defined and their behaviors are predictable.

There are several formal methods for UML, or for object-oriented design in general, but most of them are either using another high-level language or semi-formal (i.e. many parts are defined in natural language).

In this paper, we present a formal specification for the basic components in UML, including class and state diagrams. The specification is based on the basic mathematics notations like set, mapping, relation, and predicate logic, rather than relying on another high-level language that itself is subject to being defined formally. Because of the large scope of UML, we will not include other complex aspects of UML in this paper. They will be addressed in our future study.

2 Related Work

UML has become the industrial standard language for describing software design [8] after more than 10 years of evolution of the language. The most commonly used format of UML is the UML diagram. The work that had significant impact on the creation

R. Lee (Ed.): Soft. Eng., Arti. Intel., Net. & Para./Distri. Comp., SCI 149, pp. 247–257, 2008.
springerlink.com

of UML was the object-oriented concept and methodology [2, 3, 9]. There are many publications about UML, such as [6, 7]. A recent description of the usage of UML and its diagrams can be found in [12].

Formal specifications were also developed for UML or some subset of UML. For example, Damm et al. [5] developed a formal semantics for a UML kernel language. The same authors also gave a formal semantics of concurrency and communication in real-time UML [4]. Polack and Laleau [10] presented a variation of UML for the specification of the static structure of information systems. Smith proposed the Object-Z specification language [11] to describe object-oriented software components.

Although some work has been done in formal specification of UML, it is still a research area that needs a lot more attention, particularly for direct mapping of UML diagrams to formal specifications expressed in simple notations. This paper is an attempt to address this problem.

3 Basic Notations

First, let's define some basic notations.

- A *value* is given as commonly understood without further definition. Let $\mathbb{V}$ be the universal set of values, including the empty value ε.
- A *symbolic value* is a value that represents another value or a set of other values.
- A *domain* is a pair (n, D) where n is the name of the domain and D is a set of values: $D \subset \mathbb{V}$. The name of a domain is a symbolic value as an identifier of the domain and n may be ε.
- An *attribute* a denotes values drawn from given a domain. That is, an attribute is a variable that holds values from the domain: $a = v_i,\ v_i \in D$, for domain D.
- An *operation* f is a mapping

$$f : \mathbb{V} \to \mathbb{V}$$

  Or, $f(V) = V'$ where $V,\ V' \subset \mathbb{V}$. $V = V_a \cup V_p$ is a union of two subsets of values where V_a is the primary set of values and V_p is a secondary set of values. V_p is also called *parameters*. V and V' may overlap. We can think of an operation being a function applying to zero or more parameters V_p and may change the values of the parameters and may produce more values.
- An *implementation* of an operation f is the process by which the mapping of f is carried out on a machine.
- An *invocation* of an operation is an action to start the execution of an implementation of the operation.
- A *message* is an invocation of an operation f along with the secondary values V_p.
- Each attribute and operation is associated with an *access mode* that is a constant drawn from the enumeration $\{public,\ private,\ protected\}$. The semantics of access mode is defined in the next section.

4 Class and Object

There are two views of *class*, abstract-first or concrete-first. The abstract-first view is to first define a class and then define objects as instantiations of the class. On the other hand, the concrete-first view is to define objects first and then define a class as a representative of a collection of objects of the same kind. In UML diagram, classes are viewed without the associated objects and hence is an abstract-first view. Since a formal specification is given to UML classes in this paper, we also use the abstract-first view.

4.1 Class

A *class* c is a tuple $c = (n,\ m,\ A,\ F,\ p)$, where

n	*name* of c.
m	*mode* of c.
A	a finite set of attributes. A may be $\varnothing$.
F	a finite set of operations. F may be $\varnothing$.
p	a *package* the class c belongs to, i.e. $c \in p$. That is, p is a set of classes, may be $\varnothing$.

Each of the items x in the tuple c is denoted $c.x$. For example, $c.n$ denotes the name of c, $c.A$ denotes the set of attributes A of c, etc.

The name n of class c is unique in p. That is, $c_i.n \neq c_j.n,\ c_i, c_j \in p, \forall i \neq j$.

The mode m of the class is a constant from the enumeration {*interface, abstract, class*}:

interface	$\neg\exists f \in F,\ f$ is *implemented*.
abstract	$\exists f \in F,\ f$ is not *implemented*.
class	$\forall f \in F,\ f$ is *implemented*.

A class c is a domain $(c.n, D_c)$ with the domain name $c.n$ and a set of values D_c. A value in D_c consists of the values of $(A,\ F)$.

4.2 Object

An *object* o of class c is a value from the domain of the class, i.e. $o \in D_c$. The value of o contains the values of A and F. A value $a_j \in A$ in o is also called an attribute of o and is referred to as $o.a_j$, and an operation $f_k \in F$ in o is also called an operation of o and is referred to as $o.f_k$. The attribute set of o is denoted as $A(o)$ and the operation set of o is denoted as $F(o)$.

If we were to use the concrete-first view for classes, a class would be defined as $c = (n,\ m,\ O,\ p)$ where O is a set of objects of the same kind:

$$O = \{o_x \mid A(o_i) = A(o_j),\ F(o_i) = F(o_j), \forall i \neq j\}$$

Now, we can define the access mode of attributes and operations. The access mode *acc* of an attribute a_j of object o in a class c is defined as

$$acc = \begin{cases} public & o.a_j \text{ can be used in object } o' \in c' \\ & \text{for any } c' \text{ including } c' \neq c. \\ private & o.a_j \text{ can be used in object } o' \in c' \\ & \text{only if } c' = c. \\ protected & o.a_j \text{ can be used in object } o' \in c' \\ & \text{for } c' \neq c, \text{ and either } p_{c'} = p_c \text{ or} \\ & \exists s = (c',\ c,\ l) \text{ where } s \text{ is an} \\ & \text{association and } l = inheritance \end{cases}$$

Here, the phrase "$o.a_j$ can be used in object o'" means that $o.a_j$ can appear in an implementation of any operation $o'.f_k$.

The access mode of an operation f_j of object o in class c is defined the same way.

Informally, a *public* attribute or operation can be used by any other objects; *private* ones can be used only by o; and *protected* ones can be used by other objects in the package o belongs to or objects in classes c' that inherits from c. We will discuss association relationships between classes in the next section.

5 Association

Relationships between classes are represented as associations. An association may involve two or more classes. An association involving two classes are called binary association that is the one we will discuss here because it is the most important and commonly used. Multi-class associations can be derived from binary associations. A binary association is a tuple

$$S = (c_1,\ c_2,\ D)$$

where c_1, c_2 are two classes, and $D = \{d\}$ is a set of *descriptions* of the relationship between c_1 and c_2. The *description* of an association $d = (t,\ L,\ g)$ where t is an enumeration of types, L is a set of labels, and g is a directional flag. The enumeration of types includes *association, inheritance, navigability, aggregation*, *composition*, and *derived association*, etc. Note that the enumeration *association* is included here to represent a general association. The enumeration is extendable. Each label $l \in L$, if $L \neq \varnothing$, may take one of the following forms:

String	information for human users
Role	a pair $(r_1,\ r_2)$, also for human users
Multiplicity	a pair $(m_1,\ m_2)$, will be explained below.

Any of the label l can be ε. The directional flag is either *unidirectional*, represented by $c_i \rightarrow c_j$, or *bidirectional* represented by $c_i \leftrightarrow c_j$. The two classes c_i and c_j are *participants* of the association S, or they *participate* in the S, denoted as $c \prec S$.

Each m_i in multiplicity is of the form $l_i..u_i$ where $l_i, u_i \geq 0,\ l_i \leq |c_i \prec S| \leq u_i$ for $i = 1, 2$. This indicates that the degree of participation (the number of objects) of class

c_i in association S is bounded in the range $[l_i, u_i]$. The upper bound u_i may be denoted * if $u_i = \infty$.

This definition of association can be illustrated by an example. Consider the UML class diagram shown in Fig. 1.

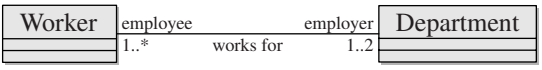


Fig. 1. UML diagram for association

The association of the two classes is specified as $(Worker, Department, D)$ for $D = \{d = (t, L, g)\}$, where

t = *association*
L = $\{works\_for, (employee, employer), (1..*, 1..2)\}$
g = *bidirectional*

Here L includes three labels: a string *works_for* and a pair of roles (*employee*, *employer*) that are for human users to understand the association, and a pair of multiplicities $(1..*, 1..2)$.

5.1 Inheritance

The *inheritance* association of class c_1 and class c_2 is unidirectional. If the two classes has an association

$$(c_1, c_2, (inheritance, L, c_2 \rightarrow c_1) \in D)$$

then the following holds:

- $c_1.A \subseteq c_2.A \ \forall i \ (a_i \in c_1.A, \ a_i.m = public)$,
- $c_1.F \subseteq c_2.F \ \forall i \ (f_i \in c_1.F, \ f_i.m = public)$,
- a_i can appear in any implementation of f, $a_i \in c_1.A$
  $f \in c_2.F$, and $a_i.m = public$ or *protected*,
- f_i can appear in any implementation of f, $f_i \in c_1.F$
  $f \in c_2.F$, and $f_i.m = public$ or *protected*.

That is, all public attributes of c_1 are also attributes of c_2, and all public operations of c_1 are also operations of c_2. An object of c_2 can directly use public or protected attributes and operations in any implementation of its operations. In the inheritance association, c_1 is said to be a *generalization* of c_2 and c_2 is said to be a *specialization* of c_1, denoted as $c_2 \lessdot c_1$.

Consider the UML diagram in Fig. 2 showing the inheritance relation `Book` $\lessdot$ `Publication`. Each of the attributes and operations is shown with an access mode: locked for private, unlocked for protected, boxed for public.

Class `Book` inherits the operation `getAuthors()` from `Publication`, in addition to its own operations. Object o of `Book` can directly use the protected attribute `title` of `Publication`.

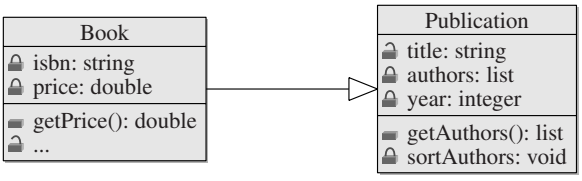


Fig. 2. UML diagram for inheritance

5.2 Aggregation

The *aggregation* association of class c_1 and class c_2 is unidirectional. If the two classes has an association

$$(c_1,\ c_2,\ (aggregation,\ L,\ c_2 \rightarrow c_1) \in D)$$

then the following holds:

$$\exists o_1.a_j(o_1 \in c_1,\ a_j = o_2 \text{ for } t_1 \leq t \leq t_2,\ o_2 \in c_2)$$

where $[t_1, t_2]$ is a time interval. In other words, object o_1 of class c_1 has an attribute a_j that holds as its value an object o_2 of class c_2 during a given time period. We say that object o_2 is part of o_1.

The example shown in Fig. 3 shows that an aggregation relation between `Course` and `Degree Program`.

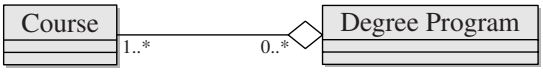


Fig. 3. UML diagram for aggregation

A `Degree Program` object contains one or more `Course` objects as its parts.

5.3 Composition

A *composition* association of class c_1 and class c_2 is an aggregation where $t_2 = t(o_2)$, the lifetime of o_2. That is, $o_1.a_j$ always hold o_2 as its value as long as o_2 is alive.

5.4 Derived Association

A *derived association* of class c_1 and c_2 is

$$(c_1,\ c_2,\ D)$$

if $\exists c_3$ such that $(c_1,\ c_3,\ D')$ and $(c_2,\ c_3,\ D'')$ are associations, $d_{13} \in D'$, $d_{23} \in D''$, and $(d_{13}, d_{23}) \Rightarrow d$ for $d \in D$. Namely, there is a class c_3 that relates to c_1 with a description d_{13} in D' and relates to c_2 with a description d_{23} in D'', and $(d_{13},\ d_{23})$ implies d.

A derived association can also be described from the object's point of view. To simplify the notation, let A be an association of two classes c_1 and c_2. We use $(o_1, o_2) \in A$ to indicate the classes c_1 and c_2 participate in A, for $o_1 \in c_1$ and $o_2 \in c_2$. A derived association can then be described as

$$(o_1,\ o_2) \in A,\ \ o_1 \in c_1,\ o_2 \in c_2,\ \text{A is a derived association}$$
$$\text{if } \exists o_3 \in c_3,\ ((o_1,\ o_3) \in A_1,\ (o_2,\ o_3) \in A_2))$$

Fig. 4 shows an example of derived association.

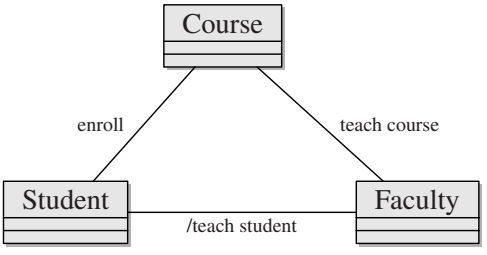


Fig. 4. UML diagram for derived association

A `faculty` *f* is associated by *teaches student* to a `Student` *s* if and only if there is some `Course` *c* that *f* is associated with *c* by *teach course* and *s* is associated with *c* by *enroll*.

5.5 Navigability

The *navigability* of class c_1 and class c_2 is bidirectional (default) or unidirectional. If two classes are related with a unidirectional navigability

$$(c_1,\ c_2,\ (\textit{navigability},\ L,\ c_1 \rightarrow c_2) \in D)$$

then $\exists o_1\ \forall f_k (o_1 \in c_1,\ f_k \in c_2.F,\ f_k.m = \textit{public},\ o_1.f_k$ can appear in an implementation of $f,\ f \in c_1.F)$. In other words, object o_2 in c_1 may use operation f_k of class c_2. Unidirectional navigability is conceptually similar to aggregation: the association is navigable from c_1 to c_2 if an object o_1 in class c_1 contains a reference via one of its attributes a_k to an object o_2 in class c_2. That is, $\exists a_k \in A(o_1)$, such that $o_1 \in c_1,\ a_k = o_2,\ o_2 \in c_2$.

This specification for unidirectional navigability can be extended for bidirectional.

5.6 Constraint

A *constraint* is an invariant on a system $(\mathscr{C},\ \mathscr{A})$ of a collection of classes $\mathscr{C}$ and associations $\mathscr{A}$. As any other invariants, a constraint can be expressed using

predict logic and set notations. The standard formal language for constraints on UML components is the Object Constraint Language (OCL) [1, 13].

5.7 Association Class

An *association class* is a class that represents an association. Let $S = (c_1,\ c_2,\ D)$ be an association of two classes c_1 and c_2. An association class representing S is defined as $c = (n,\ m,\ A,\ F,\ p)$, where

- n *name*: $n = l,\ l \in D.L,\ l$ is a string.
- m *mode*: $m = c_1.m$ or $c_2.m$ (they may be equal).
- A *attributes*: $A \subset (c_1.A \cup c_2.A)$.
- F *operations*: $f \in F,\ f$ has reference to c_1 and c_2.
- p *package*: $p = c_1.p$ or $c_2.p$ (they may be equal).

Another way to represent association S using a class c is to separate c from the association S and create two more associations $S_1 = (c, c_1, D_1)$ and $S_2 = (c, c_2, D_2)$ with D_1 and D_2 containing information from D. One piece of the information is the multiplicity. Let the multiplicity in D be $m = (m_1,\ m_2)$. The multiplicities in D_1 and D_2 are $m_{D_1} = (m_1, 1)$ and $m_{D_2} = (1, m_2)$. This is shown in an example in Fig. 5 where (a) is the original association of two classes, and class `Enroll` is introduced to represent the association shown in (b).

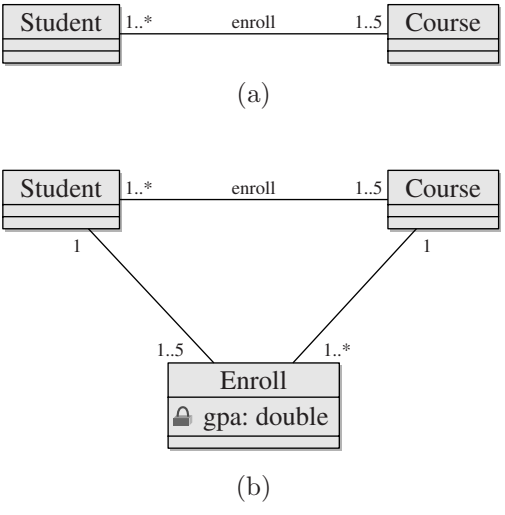


Fig. 5. UML diagram for association class

The associations specified in this section are the most fundamental and important associations. There are other types of associations but we will not include them in this paper.

6 State and Action

Once the components in classes and the relationships are specified, the next step is to specify the behavior of objects of the classes. The behavior of an object includes the activities (operations) of the object in respond to the changes in the environment.

6.1 State

A state diagram in UML describes the status of the object upon the events that occurred to the object. The events are changes in the environment (both external and internal) where the object is in. The object is informed about the changes through a message (signal or input) to the object that take the form of invocation of operations of the object.

The status of an object includes the values of its attributes and the actions the object takes in reaction to the events, as well as any output these actions may produce. It is the same concept as *finite state machine*. A state process of an object is a tuple

$$(S,\ I,\ O,\ \theta)$$

where $S = \{s\}$ is a finite set of *state*, I is a finite set of *input*, O is a finite set of *output*, and θ is a mapping

$$\theta : S \times I \rightarrow S \times O$$

θ is also called a transition function.

A state $s \in S$ is a set V of the current values of the attributes of the object x, i.e. V = values of $A(x)$. One of the states $s_0 \in S$ is the *initial state* of object x. In state s_0, all attributes $x.A$ contain their initial values. S may also contain a *final state* s_n when object x is destroyed.

I is a set of inputs (*messages*) that the object x receives. Recall in the Basic Notation section that a message is an invocation of an operation $x.f_i$ along with V_p of $x.f_i$. O is a set of outputs the object produces.

The object x changes from state s upon input i to state s' with output o under θ. That is, $\theta(s,i) = (s',o),\ s,s' \in A,\ i \in I,\ o \in O$.

An example is given in Fig. 6 showing the change of status of an object of class `Student`.

In this diagram, a `Student` object starts as an '*undergraduate*. Its state changes to "graduated" upon the message *meetDegreeRequirements()*, i.e.

$$\theta(graduated,\ meetDegreeRequirements()) = (graduated,\ \varnothing)$$

This state change does not produce output. Upon receiving the message *getJob()*, the object changes its state to *working*, and produced output *hasIncome()*:

$$\theta(graduated,\ getJob()) = (working,\ \{hasIncome()\})$$

The transition function θ is actually a sequence of one or more invocations of the operations on an object o. The state s is collective name of the values of $A(o)$, I and O are invocations of $\{f_k,\ k > 0\} \subseteq F(o)$.

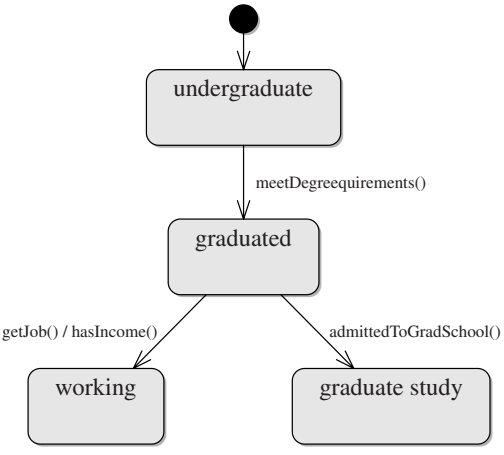


Fig. 6. State diagram for a Student object

6.2 Action

An *action* of an object is an invocation of its operations. An object takes actions in a certain order that may be sequential or parallel. UML uses action diagrams to describe the ordering. An action diagram is similar to a state diagram (or an extension of state diagram), in which a "state" is an action, and the "transition" is the ordering of the actions that may be attached with some conditions.

An action diagram is a tuple

$$(A,\ D,\ P,\ E)$$

where A is a set of *actions*, D is a set of *decisions*, P is a set of *parallel bars*, and E is a set of *directional edges*.

An action $a_i \in A$ is an invocation of an operation $f_j \in F(o)$. A decision $d_i \in D$ is a *switch* where a condition (boolean expression) is evaluated. A *parallel bar* $p_i \in P$ is a sign for synchronization of actions.

To simplify the notation, let $S = A \cup D \cup P$. The directional edge set E is a mapping

$$E:\ S \times C \rightarrow S$$

where C is a set of conditions (boolean expressions) that contains ε meaning no condition is specified. An directional edge $e_i \in E$ defines the ordering between two items in S. If $e_i = (\alpha_1,\ c) \rightarrow \alpha_2,\ \alpha_1, \alpha_2 \in S,\ c \in C$, then the following hold:

1. $t(\alpha_1) < t(\alpha_2)$ where $t(.)$ is the time function.
2. α_2 is taken only if c evaluates *true*. The condition $c = \varepsilon$ always evaluates *true*.
3. If $\alpha_2 \in P$, then $t(\alpha_k) >= t(\alpha_j), \forall k, j,\ ((\alpha_j,\ c_j) \rightarrow \alpha_2),\ (\alpha_2,\ c_k) \rightarrow \alpha_k))$. That is, any action or decision α_k (led from α_2) cannot start until all α_j (leading to α_2 has finished. Here, α_2 synchronizes the actions/decisions α_j.

7 Conclusion

Formal methods are critical to software development. As the standard specification language, UML is semi-formal and mostly expressed in graphic model as diagrams. In this paper, we provided a formal specification of the semantics of the basic components in UML, class and state diagrams, using the predicate logic and set theory.

There are other types of UML diagrams (such as sequence diagrams and use case diagrams) as well as other aspects in class and state diagrams that we did not discuss in this paper. We are currently working on the formal specifications of these UML components.

References

1. Uml 2.0 OCL specification. OMG ptc/03-10-14, OMG (2003)
2. Boehm, B.W.: A spiral model of software development and enhancement. IEEE Computer 21(5), 61–72 (1988)
3. Booch, G.: Object-Oriented Design with Applications. Ada and software engineering. Benjamin/Cummings (1991)
4. Damm, W., Josko, B., Pnueli, A., Votintseva, A.: Understanding UML: A formal semantics of concurrency and communication in real-time uml. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.-P. (eds.) FMCO 2002. LNCS, vol. 2852, pp. 71–98. Springer, Heidelberg (2003)
5. Damm, W., Josko, B., Votintseva, A., Pnueli, A.: A formal semantics for a uml kernel language, IST/33522/wp 1.1/D1.1.2, OMEGA (2003)
6. D'Souza, D., Cameron, A.: Catalysis: Objects, Frameworks and Components in UML. Addison-Wesley, Reading (1997)
7. Fowler, M., Scott, K.: UML Distilled: Applying the Standard Object Modeling Language. Addison-Wesley, Reading (1998)
8. Object Management Group. Unified modeling language: Infrustrcture, version 2.0. Technical report, OMG (2003)
9. Meyer, B.: Object-Oriented Software Construction, 2nd edn. Prentice-Hall, Englewood Cliffs (1989)
10. Polack, R., Laleau, F.: A rigorous metamodel for UML static conceptual modelling of information systems. In: Dittrich, K.R., Geppert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 71–98. Springer, Heidelberg (2001)
11. Smith, G.: The Object-Z Specification Language. Advances in Formal Methods. Kluwer Academic, Dordrecht (2000)
12. Stevens, P., Pooley, R.: Using UML Software Engineering with Objects and Components, 2nd edn. Addison-Wesley, Reading (2006)
13. Warmer, J., Kleppe, A.: The Object Constraint Language. Addison-Wesley, Reading (1999)

Author Index